

**STEGANOGRAFI METODE LEAST SIGNIFICANT BIT
PADA CITRA BITMAP DENGAN
TEKNIK KOMPRES DATA DAN EKSPANSI WADAH**



SKRIPSI

**Diajukan untuk memenuhi salah satu syarat guna mencapai gelar
Sarjana Teknik pada Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
UIN Alauddin Makassar**

Oleh :

**ABD MUIS
60200107089**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI (UIN) ALAUDDIN
MAKASSAR
2011**

ABSTRAK

Nama Penyusun : Abd Muis
Nim : 60200107089
Judul Skripsi : “Steganografi Metode Least Significant Bit Pada Citra Bitmap Dengan Teknik Kompres Data Dan Ekspansi Wadah”
Pembimbing I : Mustikasari, S.Kom, M.Kom
Pembimbing II : Abdul Wahid, S.T, M.Kom

Steganografi merupakan ilmu dan seni yang mempelajari teknik dan cara menyembunyikan pesan rahasia ke dalam suatu media sedemikian rupa sehingga pihak ketiga tidak dapat melihat dan menyadari keberadaan pesan rahasia tersebut. Tugas Akhir ini membahas studi mengenai bagaimana steganografi pada media citra digital. Citra digital yang digunakan adalah citra berformat *BMP (Bitmap)*. Steganografi pada citra *BMP*, salah satu Metode dalam Steganografi, yaitu *LSB (Least Significant Bit)*. *LSB* menambahkan bit data yang akan disembunyikan (pesan) di bit terakhir yang dimana pada proses penyisipan pesan, metode ini melakukan penyimpanan data dengan cara mengganti bit-bit yang tidak signifikan (*least significant pixel*) pada berkas (*file*) wadah (*Image*) dengan bit-bit berkas yang akan disimpan. Salah satu kelemahan dari metode modifikasi *LSB* adalah ketidakmampuannya dalam menyimpan data dengan ukuran yang besar. Untuk mengatasi hal tersebut maka dalam makalah ini dikemukakan beberapa teknik untuk memperbesar kemampuan teknik steganografi metode *LSB* dalam menyimpan data. Teknik yang pertama yaitu Teknik kompres data yaitu memperkecil ukuran file yang akan disisipkan, dan yang kedua Teknik Ekspansi wadah yaitu memperbesar wadah yang akan disisipi pesan. Dan dalam skripsi ini juga akan dilakukan analisis terhadap proses dan hasil dari masing-masing metode tersebut. Hasilnya dari percobaan ini adalah Maksimal ukuran *file* yang dapat disembunyikan adalah 850 kb pada *Image* yang mempunyai pixel 1024 x 768, sehingga besarnya ukuran pesan yang akan disisipkan tergantung dari besarnya ukuran Wadah (tempat disisipi File/data). Semakin besar ukuran pixel maka semakin besar juga ukuran pesan yang dapat disisipkan.

Kata kunci: *Steganografi, Metode LSB (Least Significant Bit), Teknik Kompres, process ekspansi wadah (Image)*.

PERSETUJUAN PEMBIMBING

Pembimbing penulisan skripsi saudara **Abd Muis**, NIM : **60200107089** , Mahasiswa Jurusan Teknik Informatika pada Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Alauddin Makassar, setelah dengan seksama meneliti dan mengoreksi skripsi yang bersangkutan dengan judul, **“Steganografi Metode *Least Significant Bit* Pada Citra *Bitmap* Dengan Teknik Kompres Data Dan Ekspansi Wadah”**, memandang bahwa skripsi tersebut telah memenuhi syarat-syarat ilmiah dan dapat disetujui untuk diajukan ke sidang *Munaqasyah*.
Demikian persetujuan ini diberikan untuk proses selanjutnya.

Makassar, 04 Agustus 2011

Pembimbing I

Pembimbing II

Mustikasari, S.Kom., M.Kom

Abdul Wahid, S.T., M.Kom

PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan di bawah ini:

Nama : Abd Muis

NIM : 60200107089

Jurusan : Teknik Informatika

Judul Skripsi : Steganografi Metode *Least Significant Bit* Pada Citra *Bitmap*
Dengan Teknik Kompres Data Dan Ekspansi Wadah.

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar merupakan hasil karya saya sendiri dan bukan merupakan pengambil alihan tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran sendiri.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut sesuai ketentuan yang berlaku.

Makassar, 10 Agustus 2010

Penyusun,

Abd Muis
NIM : 60200107089

PENGESAHAN SKRIPSI

Skripsi yang berjudul “**Steganografi Metode *Least Significant Bit* Pada Citra *Bitmap* Dengan Teknik Kompres Data Dan Ekspansi Wadah**”, yang disusun oleh Abd Muis, NIM : 60200107089, Mahasiswa Jurusan Teknik Informatika Universitas Islam Negeri (UIN) Alauddin Makassar, telah diuji dan dipertahankan dalam sidang *Munaqasyah* yang diselenggarakan pada hari Rabu, 10, Agustus 2011 M dinyatakan telah dapat diterima sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dalam Jurusan Teknik Informatika dengan beberapa perbaikan.

Gowa 10 Agustus 2011 M
10 Ramadhan 1432 H

DEWAN PENGUJI

1. Ketua : Dr. Muhammad Halifah Mustami, MPd ()
2. Sekretaris : Yusran Bobihu, S.Kom.,M.Si ()
3. Munaqisy I : Faisal Akib, S.Kom.,M,Kom ()
4. Munaqisy II : Nur Afif, S.T.,M,T ()
5. Munaqisy III : Drs. M. Arif Alim, M.Ag ()
6. Pembimbing I : Mustikasari, S.Kom.,M.Kom. ()
7. Pembimbing II : Abdul Wahid, S.T.,M.Kom ()

Diketahui oleh :

Dekan Fakultas Sains dan Teknologi
UIN Alauddin Makassar

Dr. Muhammad Halifah Mustami,M.Pd
NIP. 19711204 200003 1 001

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

KATA PENGANTAR

Alhamdulillah rabbil alamin Puji Syukur penulis panjatkan kehadiran Allah SWT yang telah memberikan Rahmat dan karunianya sehingga tugas Akhir ini dapat diselesaikan. Tugas Akhir ini disusun dan diajukan sebagai syarat untuk memperoleh sarjana pada program studi Teknik Informatika jenjang Strata-1 Universitas Islam Negeri (UIN) Alauddin Makassar.

Semoga Allah melimpahkan rahmat atas Nabi Muhammad SWA yang senangtiasa memberikan cahaya petunjuk, dan atas keluarganya yang baik dan suci dengan rahmat dan berkah-Nya menyelamatkan kita pada hari akhir.

Selama proses pembuatan Perangkat Lunak, penelitian, hingga penyusunan skripsi ini, penulis merasakan banyak hambatan dan kesulitan yang kadang membuat penulis hampir berputus asa. Namun berkat tekad dan kerja keras penulis serta dorongan dan bimbingan dari berbagai pihak, akhirnya penulis dapat menyelesaikan skripsi ini walaupun dalam bentuk yang sangat sederhana.

Atas terselesainya penulisan Skripsi ini, penulis telah mendapat banyak bantuan baik moral maupun materiil dari banyak pihak atas bantuan yang diberikan penulis tidak lupa mengucapkan Banyak Terima kasih yang sebanyak-banyaknya dan sebesar-besarnya Kepada:

1. Ayahanda Abd Rahim dan Ibunda Norma, atas segala do'a, motivasi, dan pengorbanan yang dilakukan selama penulis menyelesaikan skripsi ini. Tak akan

pernah cukup kata untuk mengungkapkan rasa terima kasih Ananda buat ayahanda dan ibunda tercinta.

2. Bapak Prof. Dr. H. A. Qadir Gassing, MS. selaku Rektor Universitas Islam Negeri (UIN) Alauddin Makassar..
3. Bapak Dr. Muhammad Halifah Mustami MPd, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Alauddin Makassar beserta staf.
4. Pembantu Dekan I, II dan III Fakultas Sains dan Teknologi
5. Bapak Faisal Akib, S.Kom., M.Kom dan Ibu Mustikasari, S.Kom., M.Kom selaku Ketua dan sekretaris Jurusan Teknik Informatika.
6. Ibu Mustikasari, S.Kom., M.Kom dan Bapak Abdul Wahid, S.T., M.Kom selaku pembimbing skripsi yang telah banyak memberikan bimbingan dan membantu penulis untuk mengembangkan pemikiran dalam penyusunan skripsi ini hingga selesai.
7. Seluruh Dosen Teknik Informatika UIN Alauddin Makassar, terima kasih atas segala ilmunya.
8. Seluruh pegawai, staf, dan karyawan Fakultas Sains dan Teknologi UIN Alauddin Makassar yang telah banyak memberikan sumbangsih baik tenaga maupun pikiran.
9. Saudaraku, Asniati, Hasnawati, dan Mustamin yang telah memberikan motivasi dan sabar menghadapi keluhan-keluhan saya selama penulisan ini.

10. My Friend's Awwal, Awaluddin, Farid, Muzakkir, Accu, Afdal, Taqim, Agus dan Ismi serta teman-teman yang lain, yang telah menjadi Teman Canda ku, dan selalu menjadi Teman Baikku, "Terima kasih Saudara".
11. Teman-temanku Teknik Informatika 2007 yang telah menjadi saudara seperjuangan menjalani suka dan duka bersama dalam menempuh pendidikan di kampus ini.

Semoga Allah swt senantiasa melimpahkan rahmat dan hidayah-Nya kepada kita semua. Seiring dengan itu pula penulis menghaturkan permohonan maaf kepada semua pihak, apabila selama proses penyusunan skripsi ini ada tutur kata tak terjaga, perilaku, dan karakter penulis yang tak terkontrol, yang tidak berkenan di hati Bapak, Ibu, dan seluruh pihak yang tidak dapat penulis sebutkan satu per satu, mohon kiranya dimaafkan karena penulis adalah manusia biasa yang tidak pernah luput dari kesalahan dan kekhilafan.

Akhir kalimat, semoga skripsi ini dapat bermanfaat bagi kita semua terlebih lagi kepada penulis sebagai penyusun.

Makassar, 10 Agustus 2011

Penulis,

Abd Muis
NIM : 60200107089

DAFTAR ISI

Halaman

HALAMAN SAMPUL.....	
HALAMAN JUDUL	
ABSTRAK	ii
HALAMAN PERSETUJUAN PEMBIMBING	iii
HALAMAN PERNYATAAN PENULIS	iv
HALAMAN PENGESAHAN.....	v
KATA PENGANTAR.....	vi
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xii
DAFTAR TABEL	xiv
BAB I PENDAHULUAN	
A. Latar Belakang	1
B. Rumusan Masalah	4
C. Batasan Masalah	5
D. Tujuan Penelitian	5
E. Manfaat Penelitian	5
F. Daftar Isi	6
BAB II LANDASAN TEORI	
A. Tinjauan Pustaka	7
B. Landasan Teori.....	9
1. Steganografi	9

2. Metode <i>Least Significant Bit (LSB)</i>	17
3. Citra Digital	20
4. Citra <i>Bitmap</i>	21
5. Compresformat ZLIB dengan Metode Lempel Ziv Welch (LZW)	27
6. Ekspansi Wadah dengan Metode Resize	28
7. Diagram Alir (Flowchart)	32
8. Unified Modelling Language (UML)	33

BAB III METODOLOGI PENELITIAN

A. Jenis Penelitian.....	37
B. Metode Pengumpulan Data	37
C. Alat Dan Bahan	38
1. Alat.....	38
2. Bahan	38
C. Lokasi Penelitian	38
D. Jadwal Penelitian.....	39

BAB IV ANALISIS DAN PERANCANGAN PERANGKAT LUNAK

A. Analisis Perangkat Lunak	40
1. Deskripsi Umum Perangkat Lunak	41
2. Alur Perangkat Lunak	42
3. Use Case Perangkat Lunak	43
B. Perancangan Sistem	46
1. Kebutuhan Sistem	47
2. Perancangan Program	47

3. Perancangan Diagram Alir Perangkat Lunak.....	47
C. Perancangan Interface	52
1. Interface Menu Beranda.....	52
2. Interface Menu Enkripsi	53
3. Interface Menu Dekripsi	55
4. Interface Menu Kompres dan Resize	56

BAB V IMPLEMENTASI DAN PENGUJIAN SISTEM

A. Implementasi Perangkat Lunak Steganografi	60
1. Implementasi Kelas.....	50
2. Implementasi Interface.....	61
B. Pengujian Sistem.....	72
1. LingkunganPengujian	70
2. Tujuan Pengujian	70
3. Data uji	71
4. Kasus uji	71
C. Analisis Dan Hasil Perangkat Lunak Steganografi.....	73

BAB VI PENUTUP

D. Kesimpulan	84
E. Saran	85

DAFTAR PUSTAKA	86
-----------------------------	-----------

DAFTAR LAMPIRAN	
------------------------------	--

DAFTAR RIWAYAT HIDUP PENULIS	
---	--

DAFTAR GAMBAR

Gambar II.1. Penyisipan pesan pada Gambar	18
Gambar II.2. Refresentasi Matriks citra.....	21
Gambar II.3. Struktur File Bitmap.....	26
Gambar II.4. Pembesaran Citra 2x2 Dengan Faktor Pembesar	29
Gambar II.5. Notasi Actor.....	34
Gambar II.6. Notasi Use Case.....	34
Gambar II.7. System Boundary	35
Gambar II.8. Association Relationship	35
Gambar III.1. Jadwal Penelitian	39
Gambar IV.1. Gambaran Umum Sistem.....	42
Gambar IV.2. Alur Perangkat Lunak	43
Gambar IV.3. Diagram Use Case	44
Gambar IV.4. Proses Penyisipan dan Pengungkapan Data.....	47
Gambar IV.5. Diagram Alir Perangkat lunak : Proses Penyisipan dan Pengungkapan, Proses Kompres dan Resize.....	50
Gambar IV.6. Proses Ubah ekstensi file ke Zlib.....	52
Gambar IV.7. Menu Utama.....	53
Gambar IV.8. Interface Menu Enkripsi Pesan	54
Gambar IV.9. Dialog untuk Proses Penyisipan Pesan	54
Gambar IV.10. Interface Menu Dekripsi Pesan	55

Gambar IV.11. Dialog Proses Pengungkapan Pesan	56
Gambar IV.12. Menu Proses Resize	57
Gambar IV.13. Dialog Proses Resize	57
Gambar IV.14. Menu Kompres.....	58
Gambar IV.15. Menu Proses Kompres File.....	59
Gambar IV.16. Menu Proses Dekompres File	59
Gambar V.1. Menu Beranda	63
Gambar V.2. Menu Proses Enkripsi Pesan	64
Gambar V.3. Dialog Memilih File.....	64
Gambar V.4. Dialog Memilih Image	65
Gambar V.5. Dialog Menyimpan gambar yang telah disisipi Pesan	65
Gambar V.6. Menu Proses Dekripsi Pesan	66
Gambar V.7. Dialog Memilih hasil Stego.....	67
Gambar V.8. Dialog Memilih Lokasi Dekripsi	67
Gambar V.9. Menu Resize.....	68
Gambar V.10. Dialog Memilih Image Resize.....	68
Gambar V.11. Dialog Menyimpan hasil Resize	69
Gambar V.12. Menu Kompres	69
Gambar V.13. Proses Kompres File	70
Gambar V.14. Proses Dekompres File.....	71
Gambar V.15. Dialog Menyimpan Hasil Kompres dan Dekompres File	71

DAFTAR TABEL

Table II.1. keterangan Gambar diagram alir Program	32
Table IV.1. Tabel Use Case	45
Table V.1. Daftar Tabel Kelas Perancangan dan Implementasi	61
Table V.2. Hasil Pengujian Kasus Uji 1	75
Table V.3. Hasil Pengujian Kasus Uji 2 (Penyisipan)	76
Table V.4. Hasil Pengujian Kasus Uji 2 (Pengungkapan)	77
Table V.5. Hasil Pengujian Kasus Uji 3 (Kompres)	77
Table V.6. Hasil Pengujian Kasus Uji 3 (Resize)	78
Table V.7. Hasil Pengujian Kasus Uji 4 Sebelum dikompres	78
Table V.8. Hasil Pengujian Kasus Uji 4 Setelah Dikompres.....	79
Table V.9. Hasil Pengujian Kasus Uji 5 Untuk menguji kualitas gambar Sebelum dan setelah disisipi Pesan	80
Table V.10. Hasil Pengujian Kasus Uji 6 Untuk Ratio Ukuran File dan gambar....	81
Table V.11. Hasil Pengujian Kasus Uji 7 Untuk pengiriman gambar yang telah disisipi pesan lewat E-mail	82

DAFTAR LAMPIRAN

```
// Satu byte untuk setiap saluran RGB, One byte for every channel of the RGB trio
type pRGBArray= ^TRGBArray;
  TRGBArray= array [1..3] of Byte;
```

```
procedure ProcessEncrypt(Bitmap: TBitmap; Source: TFileStream; Destination:
string; BPC: LongInt; ProgressBar: TProgressBar);
```

```
var SourceIndex, SourceSize: LongInt;
    BitIndex, PixelBitIndex: LongInt;
    SourceByte: Byte;
    PixelsRow: pRGBArray;
    RGBIndex: Integer;
    PixelsRowMax, PixelsRowIndex, CurrentRow: Integer;
```

//Sebuah prosedur dalam lainnya cukup jelek, tapi kami hindari melewati banyak parameter ketika kita menyebutnya

// A procedure inside another is quite ugly, but we avoid passing a lot of parameters when we call it

```
procedure CheckNextPixel;
```

```
begin
```

```
  if (RGBIndex <= 3) and (PixelBitIndex + 1 < BPC) then // We're OK, go for the
next bit Kami OK, pergi untuk bit berikutnya
```

```
    Inc(PixelBitIndex)
```

```
  else if RGBIndex < 3 then // (Beralih ke saluran RGB berikutnya )
```

```
    begin
```

```
      Inc(RGBIndex);
```

```
      PixelBitIndex:= 0;
```

```
    end
```

```
  else if RGBIndex = 3 then // (Load berikutnya pixel)
```

```
    begin
```

```
      PixelBitIndex:= 0;
```

```
      RGBIndex:= 1;
```

```
      if PixelsRowIndex = // (Kami menggunakan semua piksel dalam baris ini)
```

```
        begin
```

```
          Inc(CurrentRow);
```

```
          PixelsRow:= Bitmap.ScanLine[CurrentRow];
```

```
          PixelsRowIndex:= 1;
```

```
        end
```

```
      else // We still have pixels left in this row (Kami masih memiliki piksel tersisa di
baris ini)
```

```
        begin
```

```

    Inc(PixelsRowIndex);
    Inc(PixelsRow); // Increment the pointer so it points to the next pixel (Kenaikan
pointer sehingga menunjuk ke pixel berikutnya)
    end;
    end;
end;

```

```

begin {ProcessEncrypt}

```

```

    PixelsRow:= Bitmap.ScanLine[0];

```

```

    //Kami menggunakan 2 bit dalam saluran B untuk membawa menghitung untuk 4
bit, sehingga kita bisa menyimpan nilai BPC = 8, yang biasanya membawa
//kehancuran pada kualitas gambar

```

```

    SetBitAt(PixelsRow^[1], 0, GetBitAt(BPC, 0));
    SetBitAt(PixelsRow^[1], 1, GetBitAt(BPC, 1));
    SetBitAt(PixelsRow^[2], 0, GetBitAt(BPC, 2));
    SetBitAt(PixelsRow^[3], 0, GetBitAt(BPC, 3));

```

```

    // Initialize

```

```

    PixelsRowMax:= Bitmap.Width;
    CurrentRow:= 0;
    PixelBitIndex:= 0;
    PixelsRowIndex:= 2;
    RGBIndex:= 1;
    Inc(PixelsRow);

```

```

///STORE YANG PANJANG AKTUAL DATA

```

```

    SourceSize:= Source.Size;
    for BitIndex:= 0 to SizeOf(SourceSize) * 8 - 1 do
    begin
        SetBitAt(PixelsRow^[RGBIndex], PixelBitIndex, GetBitAt(SourceSize,
BitIndex));
        CheckNextPixel;
    end;

```

```

    {-- STORE THE DATA --}

```

```

    //Salin bit dari setiap byte dalam aliran sumber ke bit-bit pada piksel
    Source.Seek(0, soFromBeginning);
    for SourceIndex:= 0 to SourceSize - 1 do
    begin

```

```

        if (SourceIndex + 1) mod 10 = 0 then

```



```

begin
    ProgressBar.StepIt;
    Application.ProcessMessages;
end;

Source.Read(SourceByte, 1);
for BitIndex:= 0 to 7 do
begin
    SetBitAt(PixelsRow^[RGBIndex], PixelBitIndex, GetBitAt(SourceByte,
BitIndex));
    CheckNextPixel;
end;

end;// for SourceIndex
end;

procedure Encrypt(const SourceFile, SourceBitmap, Destination: string;
BitsPerChannel: LongInt; ProgressBar: TProgressBar);
var Bitmap: TBitmap;
    sSource: TFileStream;

begin
    ProgressBar.Position:= 0;
    ProgressBar.Step:= 1;

    Bitmap:= TBitmap.Create;
    sSource:= nil;
    try
        Bitmap.LoadFromFile(SourceBitmap);

        if Bitmap.PixelFormat <> pf24bit then
            raise Exception.Create('Gambar harus memiliki kedalaman 24-bit');

        sSource:= TFileStream.Create(SourceFile, fmOpenRead);

        if sSource.Size = 0 then
            raise Exception.Create('The source file is 0 bytes. There"s nothing to hide (File
sumber 0 byte. Tidak ada yang disembunyikan).');

        if sSource.Size * 8 + SizeOf(LongInt) * 8 + 3 > Bitmap.Width * Bitmap.Height *
3 * BitsPerChannel then
            raise Exception.Create('The image is not big enough to accommodate the
file(Gambar tidak cukup besar untuk menampung file).');
        // (+ 1: Nilai BitsPerChannel disimpan dalam pixel pertama)

```

```

    // ( * 8: kami akan menyimpan jumlah data dienkripsi aktual dalam piksel
//pertama setelah BitsPerChannel

    ProgressBar.Max:= sSource.Size div 10;

    ProcessEncrypt(Bitmap, sSource, Destination, BitsPerChannel, ProgressBar);

    Bitmap.SaveToFile(Destination);

finally
    Bitmap.Free;
    if Assigned(sSource) then sSource.Free;
end;

end;

procedure ProcessDecrypt(Bitmap: TBitmap; Destination: TFileStream; ProgressBar:
TProgressBar);
var DataSize, DataIndex: LongInt;
    Data, BitIndex: Byte;
    PixelsRow: pRGBArray;
    PixelsRowMax, PixelsRowIndex, CurrentRow, MaxRows: Integer;
    PixelBitIndex: LongInt;
    RGBIndex: Integer;
    BPC: LongInt;

//Sebuah prosedur dalam lainnya (dan pada dasarnya sama dengan yang di
//banyak parameter ketika kita menyebutnya
procedure CheckNextPixel;
begin
    if (RGBIndex <= 3) and (PixelBitIndex + 1 < BPC) then // We're OK, go for the
next bit ( OK, ke bit berikutnya)
        Inc(PixelBitIndex)
    else if RGBIndex < 3 then //( beralih kesaluran RGB berikutnya)
        begin
            Inc(RGBIndex);
            PixelBitIndex:= 0;
        end
    else if RGBIndex = 3 then
        begin
            PixelBitIndex:= 0;
            RGBIndex:= 1;
            if PixelsRowIndex = PixelsRowMax then
                begin
                    Inc(CurrentRow);

```

```

    if CurrentRow > MaxRows then
        raise Exception.Create(' Akhir gambar itu tiba saat mencoba membaca informasi
yang tersembunyi. "+ # 13 # 10 +
        'Ini mungkin disebabkan oleh sebuah gambar yang tidak
mengandung data tersembunyi. ');

```

```

    PixelsRow:= Bitmap.ScanLine[CurrentRow];
    PixelsRowIndex:= 1;
    end
    else //(Kami masih memiliki piksel tersisa di baris ini)
    begin
        Inc(PixelsRowIndex);
        Inc(PixelsRow);
        end;      //Kenaikan pointer sehingga menunjuk ke pixel berikutnya
    end;
end;

```

```

begin {ProcessDecrypt}

```

```

    //DAPATKAN BITS PER NILAI CHANNEL

```

```

    PixelsRow:= Bitmap.ScanLine[0];
    BPC:= 0;
    SetBitAt(BPC, 0, GetBitAt(PixelsRow^[1], 0));
    SetBitAt(BPC, 1, GetBitAt(PixelsRow^[1], 1));
    SetBitAt(BPC, 2, GetBitAt(PixelsRow^[2], 0));
    SetBitAt(BPC, 3, GetBitAt(PixelsRow^[3], 0));

```

```

    if (BPC < 1 ) or (BPC > 8) then
        raise Exception.Create(' Nilai BitsPerChannel tidak dalam kisaran 1-8. "+ # 13 #
10 +
        'Ini mungkin disebabkan oleh sebuah gambar yang tidak mengandung data
tersembunyi ');

```

```

        // Initialize

```

```

    PixelsRowMax:= Bitmap.Width;
    MaxRows:= Bitmap.Height - 1;
    CurrentRow:= 0;
    PixelBitIndex:= 0;
    PixelsRowIndex:= 2;
    RGBIndex:= 1;
    Inc(PixelsRow);

```

```

    {-- DATA HIDDEN --}

```

```

for BitIndex:= 0 to SizeOf(DataSize) * 8 - 1 do
begin
  SetBitAt(DataSize, BitIndex, GetBitAt(PixelsRow^[RGBIndex], PixelBitIndex));
  CheckNextPixel;
end;

if DataSize <= 0 then
  raise Exception.Create(' Ukuran disimpan dari data tersembunyi adalah tidak
benar. "+ # 13 # 10 +
'Ini mungkin disebabkan oleh sebuah gambar yang tidak mengandung data
tersembunyi.');
```

ProgressBar.Max:= DataSize div 10;

{-- EXTRACT DATA AKTUAL --}

```

for dataIndex:= 1 to DataSize do
begin

  if dataIndex mod 10 = 0 then
  begin
    ProgressBar.StepIt;
    Application.ProcessMessages;
  end;

  for BitIndex:= 0 to 7 do
  begin
    SetBitAt(Data, BitIndex, GetBitAt(PixelsRow^[RGBIndex], PixelBitIndex));
    CheckNextPixel;
  end;

  Destination.Write(Data, 1);

end; //for dataIndex
end;
```

procedure Decrypt(const SourceFile, DestFile: string; ProgressBar: TProgressBar);
var Bitmap: TBitmap;
 Destination: TFileStream;

```

begin
  ProgressBar.Position:= 0;
  ProgressBar.Step:= 1;
```

```

Bitmap:= TBitmap.Create;
Destination:= nil;
try

try
    if FileExists(DestFile) then DeleteFile(PAnsiChar(DestFile));

    Destination:= TFileStream.Create(DestFile, fmCreate);
    Bitmap.LoadFromFile(SourceFile);

    if Bitmap.PixelFormat <> pf24bit then
        raise Exception.Create("The image doesn't have a 24-bit depth. It surely hasn't
been created by this program.");
        //Gambar tidak memiliki kedalaman 24-bit. Ini pasti belum
diciptakan oleh program ini.
    ProcessDecrypt(Bitmap, Destination, ProgressBar);

finally
    Bitmap.Free;
    if Assigned(Destination) then Destination.Free;
end;

except
    if FileExists(DestFile) then
        DeleteFile(PChar(DestFile));
    raise;
end;
end;

end.

```

RIWAYAT HIDUP PENULIS



ABD MUIS, lahir di Kabupaten Barru pada tanggal 05 Mei 1987. Anak dari pasangan suami istri Bapak Abd Rahim dan Ibu Norma yang merupakan anak bungsu dari empat bersaudara. Memulai pendidikannya di SD Inpres Watu pada tahun 1994 – 1998 selama lima tahun. Tahun 2000 – 2003 di SLTP Negeri 5 Watu, 2003 – 2006 di SMK

Teknologi Pembangunan Barru dan kemudian 2006 melanjutkan Politeknik Negeri Pangkep, jurusan Agribisnis dan pada tahun 2007-2010 ke Universitas Islam Negeri (UIN) Alauddin Makassar, jurusan Teknik Informatika.

Selama kuliah di UIN Alauddin, Kegiatan di luar kampus adalah sebagai, anggota FORMAB (Forum Mahasiswa Barru) dan anggota Study Club Explorasi Solidaritas Mahasiswa Teknik Informatika (EXOMATIK). Pernah mengajar di JILC sebagai tentor tahun 2009 - 2011.



BAB I

PENDAHULUAN

A. Latar Belakang

Keamanan merupakan karunia Allah yang diberikan kepada manusia yang wajib kita syukuri. Allah juga yang telah mengaruniakan manusia akal untuk berpikir. Masalah keamanan merupakan hal yang terpenting dalam kehidupan didunia ini. Oleh karena itu manusia wajib menggunakan akalnya untuk mempelajari dan menciptakan keamanan itu.

Sebagaimana Firman Allah SWT dalam Al-Qur'an Surah Al Hasyr : 23:

هُوَ اللَّهُ الَّذِي لَا إِلَهَ إِلَّا هُوَ الْمَلِكُ الْقُدُّوسُ السَّلَامُ الْمُؤْمِنُ الْمُهَيْمِنُ الْعَزِيزُ الْجَبَّارُ
الْمُتَكَبِّرُ سُبْحَنَ اللَّهِ عَمَّا يُشْرِكُونَ ﴿٢٣﴾

Terjemahannya :

*Dialah Allah yang tiada Tuhan selain Dia, raja, yang Maha suci, yang Maha Sejahtera, yang Mengkaruniakan Keamanan, yang Maha Memelihara keimanan, yang Maha Perkasa, yang Maha Kuasa, yang memiliki segala Keagungan, Maha Suci Allah dari apa yang mereka persekutukan.*¹

Dari ayat al qur'an diatas disebutkan bahwa Allah adalah Tuhan yang wajib kita sembah. Allah yang Maha memelihara, Yang Maha perkasa, Yang Maha Kuasa, Yang memiliki segala Keagungan, juga Yang mengaruniakan keamanan bagi manusia.

¹ Departemen Agama RI, *Alquran dan Terjemahannya* (Jakarta : CV Toha Putra Semarang, 1989), h. 915.

Dewasa ini kebutuhan manusia akan informasi semakin meningkat. Terlebih lagi dengan perkembangan Teknologi informasi yang semakin pesat dan semakin cepat membuat pertukaran informasi menjadi lebih mudah dan cepat, pertukaran informasi dapat dilakukan melalui berbagai macam media salah satunya adalah melalui Media digital. Penggunaan media digital juga semakin meningkat. Populernya penggunaan media digital sebagai media pertukaran informasi disebabkan karena kemudahan yang ditawarkan media digital kepada para penggunanya. Namun sifat dari media digital ini mempunyai kelemahan untuk kasus pertukaran informasi yang bersifat rahasia. Pencurian informasi sering terjadi di dunia internet, sehingga hal ini membuat pertukaran informasi yang bersifat rahasia harus dilakukan dengan hati-hati. Selain itu media digital dapat secara cepat tersebar melalui jaringan internet. seiring berkembangnya teknologi informasi tersebut semakin berkembang pula kejahatan sistem informasi. Dengan berbagai teknik banyak yang mencoba untuk mengakses informasi yang bukan haknya. maka dari itu sejalan dengan berkembangnya teknologi informasi harus juga dibarengi dengan perkembangan pengamanan sistem informasi.

Sehubungan dengan surah *Al-Hasyr* yang mengemukakan pentingnya keamanan dalam hal pertukaran informasi sehingga Alternative keamanan yang dapat menangani kerahasiaan informasi itu, dikenal dengan Teknik Kriptografi.

Kriptografi adalah ilmu sekaligus seni untuk menjaga keamanan pesan. Kriptografi mengubah informasi asli (*Plaintext*) melalui proses enkripsi menjadi informasi acak (*chipertext*) menggunakan algoritma dan kunci tertentu, lalu setelah diterima oleh penerima informasi, *Chipertext* akan diubah menjadi *Plaintext* melalui proses deskripsi menggunakan algoritma dan kunci yang sama dengan proses enkripsi sehingga pesan rahasia hanya dapat dimengerti oleh pihak penerima.²

Namun Kriptografi juga mempunyai kelemahan. informasi acak yang dikirim menggunakan algoritma dan kunci tertentu berbentuk sebuah pesan yang tidak mempunyai makna, namun dapat dilihat secara kasat mata. Hal ini dapat menimbulkan kecurigaan atau mudah dideteksi oleh pihak-pihak yang tidak bertanggung jawab. Oleh karena itu dipakai teknik lain untuk menjawab kelemahan teknik Kriptografi. Teknik ini dinamakan teknik Steganografi. Steganografi adalah ilmu dan seni menyembunyikan informasi yang dapat mencegah pendeteksian terhadap informasi yang disembunyikan. Pada steganografi informasi asli langsung disisipkan pada media lain (*Cover-Image*), lalu media yang telah disisipkan informasi (*Stego-Image*) tadi dapat dipertukarkan kepada penerima. Steganografi mempunyai keunggulan yaitu tidak ada perbedaan secara kasat mata antara *Cover-Image* dengan *Stego-Image*. Media yang dapat disisipkan oleh informasi rahasia dapat berupa teks, citra, audio maupun video. Jumlah pertukaran data media besar, membuat kemungkinan kecurigaan adanya

² Rinaldi Munir, *Kriptografi* (Bandung : Informatika Bandung, 2006), h. 301.

informasi rahasia yang pertukarkan melalui perukaran media digital menjadi kecil.

Jadi Perangkat yang akan dibangun ini adalah steganografi dengan memanfaatkan Metode LSB (*Least Significant bit*). *Least Significant Bit* adalah posisi bit pada bilangan biner yang memberikan nilai unit yaitu menentukan apakah nomor genap atau ganjil. Adapun kelemahan yang dimiliki oleh metode modifikasi LSB hanya mampu menyimpan data berukuran seperdelapan dari ukuran wadah atau sama dengan wadah berupa citra 24-bit, tentu saja hal ini tidak efisien. Untuk mengatasi hal tersebut maka dalam skripsi ini dikemukakan beberapa metode untuk memperbesar kemampuan teknik steganografi metode modifikasi LSB dalam menyimpan data. Metode yang pertama adalah melakukan teknik kompres terhadap data yang akan disimpan, kedua adalah ekspansi wadah yaitu melakukan proses terhadap wadah (*cover*) dengan cara memperbesar wadah.

B. Rumusan Masalah

Berdasarkan latar belakang yang telah dikemukakan di atas maka disusun rumusan masalah yang akan dibahas pada tugas Akhir ini yakni bagaimana membangun aplikasi Steganografi pada *file* citra digital berformat BMP (*Bitmap*) dengan Metode LSB (*Least Significant Bit*) dengan Teknik kompres dan Ekspansi wadah.

C. Batasan Masalah

Batasan Masalah pada Tugas Akhir ini adalah

1. Format citra digital yang digunakan adalah Format BMP (*Bitmap*) yang digunakan adalah citra berwarna. citra BMP yang setiap pixel terdiri dari 24 bit.
2. Jumlah minimal *pixel* Gambar yang digunakan disini adalah 1024 x 768 size
3. File-file atau data-data yang Disisipkan harus memiliki ekstensi sebagai berikut *.bmp, *.doc, *.pdf, *.txt, *.ico, *.gif, *.jpg, *.ppt, *.Xlsx dan *.exe.

D. Tujuan dan Manfaat Penelitian

1. Tujuan Penelitian

Tujuan penelitian ini adalah untuk membangun perangkat lunak yang mengimplementasikan program penyisipan pesan pada citra digital berformat BMP dengan metode LSB (*Least Significant Bit*).

2. Manfaat Penelitian

Manfaat yang ingin dicapai dalam Penelitian ini adalah :

- a. Bagi dunia akademik sebagai referensi pengetahuan tentang teknik Steganografi.
- b. Bagi masyarakat memberi referensi dan alternative untuk metode pengiriman pesan rahasia yang aman.

E. Daftar Isi

- BAB I** Bagian pendahuluan meliputi latar belakang, rumusan masalah, batasan masalah, pengertian judul, tujuan dan manfaat penelitian, dan sistematika penulisan.
- BAB II** Bagian Kajian Pustaka meliputi Tinjauan pustaka dan Landasan teori yang akan menguraikan konsep-konsep yang mendukung dan mendasari pelaksanaan tugas akhir ini, meliputi Steganografi, Metode LSB, Citra Digital, Citra BMP, Compress format ZLIB dengan Metode Lempel Ziv Welch (LZW), Ekspansi wadah dengan Metode Resize, Diagram Alir (*Flowchart*), Unified Modelling Language (UML).
- BAB III** Bagian metode penelitian meliputi jenis penelitian, metode pengumpulan data, alat dan bahan, urutan kegiatan dan jadwal penelitian
- BAB IV** Bagian Analisis dan Perancangan Perangkat Lunak meliputi Analisis Perangkat Lunak, Perancangan Sistem dan Perancangan Interface.
- BAB V** Bagian Implementasi dan pengujian sistem meliputi Implementasi Perangkat Lunak Steganografi serta Pengujian Sistem Analisis dan Hasil Perangkat Lunak Steganografi.
- BAB VI** Bagian penutup meliputi kesimpulan dan saran mengenai tugas akhir.

BAB II

LANDASAN TEORI

Implementasi steganografi dengan memanfaatkan metode LSB (*Least Significant Bit*) bukan merupakan pembahasan yang baru di Indonesia sebelumnya materi tersebut telah diteliti tetapi dengan menggunakan beberapa metode, salah satunya adalah *Algoritma GifShuffle* dengan media implementasi steganografi yang berbeda.³ Pada bab ini jelaskan mengenai tinjauan pustaka dan landasan teori yang berkaitan erat dengan perangkat lunak yang dibangun.

A. Tinjauan Pustaka

Sebelumnya, Sulhasni Burhanuddin telah membahas perangkat lunak yang dikembangkan pada lingkungan perangkat *mobile phone*. Yaitu membangun aplikasi steganografi pada *file* citra digital berformat *Bitmap* dengan metode *DCT Modification* dan mengimplementasikannya pada *mobile phone*. Kualitas *Image* ini bergantung pada besar kecilnya pesan yang disisipkan.

Kemudian Muhammad Hakim A telah membahas tentang Studi dan Implementasi Penyembuyian Pesan dengan Metode *Least Significant Bit*, dimana Metode yang dipakai sama cara pemakaiannya dan kelemahan yang ada pada

³ Ronald Augustinus Penalosa, *Steganografi Pada Citra dengan Format GIF Menggunakan Algoritma GifShuffle* (Bandung: ITB, 2008).

Metode yang dipakai tersebut yaitu terbatasnya modifikasi LSB hanya mampu menyimpan data berukuran seperdelapan dari ukuran wadah.⁴

Kemudian Prasetyo Andy Wicaksono telah membahas tentang penyembunyian pesan pada citra GIF dengan menggunakan metode adaptif. Dimana penyembunyian *Image* ini menggunakan citra GIF dengan color image. Kualitas citra GIF dihasilkan bergantung pada besarnya ukuran Pesan. Citra GIF yang disisipkan dengan ukuran pesan yang lebih besar akan mengalami perubahan kualitas yang lebih besar.⁵

Kemudian Winda Winanti, membahas tentang penyembunyian pesan pada citra terkompresi JPEG menggunakan Metode *Spread Spectrum*. Dengan metode ini pesan dikodekan dan disebar ke setiap spectrum frekuensi yang memungkinkan. Metode ini mentransmisikan sebuah sinyal pita informasi yang sempit ke sebuah kanal pita lebar dengan penyebaran frekuensi. Kualitas Citra terkompresi JPEG yang dihasilkan bergantung dari besarnya ukuran pesan.⁶

Kemudian Ferry Pangaribuan membahas tentang penyembunyian pesan dengan metode Mars dan Zhang. Sejumlah perubahan representasi data diperlukan untuk membangun sistem kombinasi ini. Algoritma Zhang yang operasi internalnya yang berorientasi word, sedangkan algoritma Zhang yang

⁴ Muhammad Hakim, *Implementasi Penyembunyian pesan dengan metode LSB* (Bandung : Institut Teknologi Bandung, 2009).h.1.

⁵ Prasetyo Andy Wicaksono, *Penyembunyian Pesan pada Citra GIF Menggunakan Metode Adaptif* (Bandung : Institut Teknologi Bandung, 2009) hII-2.

⁶ Winda Winanti *Penyembunyian pesan pada citra terkompresi JPEG menggunakan metode Spread Spectrum* (Bandung : Institut Teknologi Bandung, 2009). h.1.

operasi internalnya berorientasi bit dan byte; sehingga perubahan representasi perlu dikendalikan agar memberikan hasil yang diharapkan. penentuan parameter ukuran arsip pesan dan ukuran citra akan berpengaruh pada performansi CombinoZM.⁷

Perbedaan perangkat lunak penyembunyian pesan yang terdapat diatas adalah Steganografi yang penulis buat berfokus pada proses Kompres dengan format ZLIB dan juga ekspansi wadah dengan menggunakan Algoritma Resize Bitmap. dengan memanfaatkan metode LSB (*Least Significant Bit*) dan format citra yang digunakan adalah format BMP (*Bitmap*).

B. Landasan Teori

1. Steganografi

Steganografi adalah seni dan ilmu menulis atau menyembunyikan pesan tersembunyi dengan suatu cara sehingga selain si pengirim dan si penerima, tidak ada seorangpun yang mengetahui atau menyadari bahwa ada suatu pesan rahasia. Kata steganografi berasal dari bahasa Yunani *steganos*, yang artinya “tersembunyi atau terselubung”, dan *graphein*, “menulis”.⁸

Herodotus adalah seorang sejarawan Yunani pertama yang menulis tentang steganografi, yaitu ketika seorang raja kejam Yunani bernama Histaeus dipenjarakan oleh Raja Darius di Susa pada abad ke-5 sebelum

⁷ Ferry Pangaribuan, *Aplikasi Penyembunyian Pesan Metode MARS Metode dan Zhang LSB Image*, (Bandung : Institut Teknologi Bandung, 2008). h.1.

⁸ Morkel, T., Eloff, J.H.P., Olivier, M.S. (2005). An Overview of Citra Steganografi

Masehi. Histaeus harus mengirim pesan rahasia kepada anak laki-lakinya, Aristagoras di Militus. Ia menulis pesan dengan cara menato pesan pada kulit kepala seorang budak. Ketika rambut budak itu mulai tumbuh, Histaeus mengutus budak itu ke Militus untuk mengirim pesan dikulit kepalanya tersebut kepada Aristagoras.

Cerita lain yang ditulis oleh herodotus, yaitu Demeratus, mengisahkan seorang Yunani yang akan mengabarkan berita kepada Sparta bahwa Xerxes bermaksud menyerbu Yunani. Agar tidak diketahui pihak Xerxes, Demaratus menulis pesan dengan cara mengisi tabung kayu dengan lilin dan menulis pesan dengan cara mengukirnya pada bagian bawah kayu. Papan kayu tersebut dimasukkan kedalam tabung kayu, kemudian tabung kayu ditutup kembali dengan lilin.

Teknik steganografi yang lain adalah tinta yang tak terlihat. Teknik ini kali pertama digunakan pada zaman Romawi kuno, yaitu dengan menggunakan air sari buah jeruk, urin, atau susu sebagai tinta untuk menulis pesan. Cara membacanya adalah dengan dipanaskan diatas nyala lilin. Tinta yang sebelumnya tidak terlihat, ketika terkena panas akan berangsur-angsur menjadi gelap sehingga pesan dapat dibaca. Teknik ini juga pernah digunakan pada Perang Dunia II.

Pada masa lampau steganografi sudah dipakai untuk berbagai kebutuhan, seperti kepentingan politik, militer diplomatik, serta untuk

kepentingan pribadi, yaitu alat komunikasi pribadi. Beberapa penggunaan steganografi pada masa lampau bisa kita lihat dalam beberapa peristiwa berikut ini :

- a. Pada Perang Dunia II, Jerman menggunakan microdots untuk berkomunikasi. Penggunaan teknik ini biasa digunakan pada microfilm chip yang harus diperbesar sekitar 200 kali. Dalam hal ini Jerman menggunakan steganografi untuk kebutuhan perang sehingga pesan rahasia strategi atau apapun tidak bisa diketahui oleh pihak lawan. Teknologi yang digunakan dalam hal ini adalah teknologi baru yang pada saat itu belum bisa digunakan oleh pihak lawan.
- b. Pada Perang Dunia II, Amerika Serikat menggunakan suku Indian Navajo sebagai media untuk berkomunikasi. Dalam hal ini Amerika Serikat menggunakan teknologi kebudayaan sebagai suatu alat dalam steganografi. Teknologi kebudayaan ini tidak diketahui atau dimiliki pihak lawan, kecuali oleh Amerika Serikat.

Dari catatan sejarah dan contoh-contoh steganografi konvensional tersebut, kita dapat melihat bahwa semua teknik steganografi konvensional selalu berusaha merahasiakan pesan dengan cara menyembunyikan, mengamufase, ataupun menyamarkan pesan.

Sementara, saat ini perkembangan teknologi internet telah membawa perubahan besar bagi kecepatan pertukaran informasi maupun distribusi

media digital. Media digital berupa teks, citra, audio, atau video dapat dipertukarkan atau didistribusikan dengan mudah melalui internet. Disisi lain, kemudahan ini dapat menimbulkan permasalahan ketika media tersebut adalah media yang sifatnya rahasia. Masalah ini juga bisa terjadi jika media tersebut terlindungi oleh hak cipta (*copyright*), tetapi dengan mudah orang lain membuat salinan yang sulit dibedakan dengan aslinya dan dengan mudah pula salinan tersebut didistribusikan atau diperbanyak oleh pihak-pihak yang tidak berhak.

Sejak 1 Januari 2000 Indonesia dan Negara anggota World Trade Organization telah menerapkan perlindungan Hak Atas Kekayaan Intelektual (HAKI). Indonesia juga termasuk salah satu negara penanda tangan persetujuan TRIPs (Trade Related Aspects of Intellectual Property Rights) pada 1994. Namun demikian, di Indonesia tetap saja banyak beredar barang-barang bajakan, berupa *compact disc* (baik berisi program aplikasi kantor, permainan, lagu, film, dan sebagainya), kaset audio, dan media elektronik lain. Barang-barang bajakan ini telah banyak digunakan sebagai media pendistribusi yang berisi informasi, khususnya yang diperoleh dari penyadapan saluran komunikasi data melalui internet.

Hal inilah yang mengharuskan orang membuat metode untuk melindungi hak cipta pada media digital. Banyak teknik yang telah dikembangkan untuk kebutuhan proteksi media digital, antara lain

Kriptografi, Steganografi, Watermarking. Pada prinsipnya ketiga teknik tersebut bisa diterapkan pada media teks, citra, audio, dan video.

Beberapa Implementasi lain dari aplikasi steganografi selain sebagai alternatif untuk menangani penyalahgunaan pasal 41 UU no. 36 tahun 1999 yang telah dipaparkan pada latar belakang, antara lain :

a. Penyimpanan data penting

Kasus penyebaran data pribadi akibat hilangnya perangkat keras, seperti *nootbook*, *harddisk*, dan media penyimpanan lainnya yang marak terjadi di masyarakat sekarang ini menambah kebutuhan akan adanya aplikasi yang dapat digunakan untuk menyimpan data dengan aman, sebagai alternatif untuk pencegahan penyalahgunaan data pribadi akibat hilangnya perangkat keras tersebut dapat digunakan aplikasi steganografi sebagai penyimpanan data penting. Dimana data penting yang dapat disimpan antar lain : pesan rahasia, nomor pin, dan nomor-nomor penting lainnya tanpa menimbulkan kecurigaan.

b. Melindungi Hak cipta

Di dunia digital, steganografi muncul dalam bentuk digital watermark, yaitu tanda digital yang disisipkan dalam gambar (*digital image*) atau suara. Hak cipta (*copyright*) dari gambar dapat disisipkan dengan menggunakan high-bit dari *pixel* yang membentuk gambar tersebut. Pada steganografi gambar yang telah disisipkan pesan terlihat

tidak berbeda, karena kemampuan (atau lebih tepatnya ketidakmampuan) mata manusia yang tidak dapat membedakan satu bit saja, akan tetapi sebenarnya gambar tersebut mengandung pesan-pesan tertentu.

c. *Belanja On-Line*

Dewasa ini banyak aplikasi-aplikasi berbasis web yang dibuat termasuk aplikasi penjualan. Sangat besar kemungkinan aplikasi penjualan online didalam meningkatkan jumlah penjualan dan mempermudah pembeli untuk membeli barang. Tingkat keamanan dari sistem penjualan online masih diragukan oleh banyak orang khususnya di Indonesia. Melihat banyaknya kasus penyadapan informasi-informasi melalui internet termasuk penyadapan nomor kartu kredit pembeli. Faktor keamanan dari sistem penjualan online perlu didesain dengan baik. Alternatif pengamanan nomor kartu kredit yang dapat digunakan adalah dengan menggunakan aplikasi steganografi, yakni dengan cara menyimpannya kedalam gambar, baru kemudian dikirimkan.

Kriptografi yang merupakan induk dari steganografi, melakukan pengubahan pada informasi rahasia menggunakan suatu algoritma tertentu yang sudah disepakati bersama antara kedua belah pihak yang akan bertukar informasi rahasia, dan ketika informasi acak ini sudah sampai pada pihak penerima, langkah terakhir dari teknik ini adalah mengubah kembali pesan itu dengan algoritma yang sama untuk mendapatkan pesan yang asli. Andaikata

terdapat pihak ketiga yang ingin menyadap dan mengambil pesan yang dikirim, pihak tersebut hanya akan mendapatkan pesan acak yang sulit dimengerti isinya.

Teknik steganografi meliputi banyak sekali metode komunikasi untuk menyembunyikan pesan rahasia (teks atau gambar) di dalam *file-file* lain yang mengandung teks, *image*, bahkan audio tanpa menunjukkan ciri-ciri perubahan yang nyata atau terlihat dalam kualitas dan struktur dari *file* semula. Tujuan dari steganografi adalah merahasiakan atau menyembunyikan keberadaan dari sebuah pesan tersembunyi atau sebuah informasi.

Dalam praktek penggunaan steganografi kebanyakan diselesaikan dengan membuat perubahan tipis terhadap data digital lain yang isinya tidak akan menarik perhatian dari penyerang potensial, sebagai contoh sebuah gambar yang terlihat tidak berbahaya. Perubahan ini bergantung pada kunci (sama pada kriptografi) dan pesan untuk disembunyikan. Orang yang menerima gambar kemudian dapat menyimpulkan informasi terselubung dengan cara mengganti kunci yang benar ke dalam algoritma yang digunakan.

Kelebihan steganografi yang dikembangkan dari kriptografi adalah pesan-pesannya tidak menarik perhatian orang lain. Pesan-pesan berkode dalam kriptografi yang tidak disembunyikan, walaupun tidak dapat dipecahkan, akan menimbulkan kecurigaan. Untuk mendapatkan hasil yang lebih baik maka digunakan steganografi untuk menjamin keamanan pesan

rahasianya. Sebuah pesan steganografi (*plaintext*), biasanya pertama-tama dienkripsikan dengan beberapa arti tradisional, yang menghasilkan *ciphertext*. Kemudian, *covertext* dimodifikasi dalam beberapa cara sehingga berisi *ciphertext*, yang menghasilkan *stegotext*. Contohnya, ukuran huruf, ukuran spasi, jenis huruf, atau karakteristik *covertext* lainnya dapat dimanipulasi untuk membawa pesan tersembunyi, hanya penerima (yang harus mengetahui teknik yang digunakan) dapat membuka pesan dan mendekripsikannya.

Kebanyakan algoritma steganografi menggunakan sebuah kombinasi dari bidang jenis teknik untuk melakukan sebuah tugas dalam penyelubungan pesan rahasia dalam sebuah selubung *file*.

Sebuah program steganografi dibutuhkan untuk melakukan hal-hal berikut (baik implisit melalui suatu perkiraan maupun eksplisit melalui sebuah perhitungan), menemukan kelebihan bits dalam selubung *file* yang dapat digunakan untuk menyelubungi pesan rahasia didalamnya, memilih beberapa diantaranya untuk digunakan dalam menyelubungi data dan penyelubungan data dalam bits dipilih sebelumnya.

Teknik penyisipan data kedalam *cover-object* dapat dilakukan dalam dua ranah :⁹

⁹ Yulie Anniera Sinaga. *Program Steganalisis Metode LSB pada Citra dengan Enhanced LSB, Uji Chi-Square, dan RS-Analysis*, (Bandung : Institut Teknologi Bandung, 2009).h.II.4.

a. Ranah Spasial

Tenkin ini mengubah langsung nilai byte dari *cover-object* (nilai byte dapat merepresentasikan intensitas/warna *pixel* atau amplitudo). Contoh metode yang tergolong dalam teknik ini adalah metode LSB (*Least Significant Bit*).

b. Ranah Transform

Teknik ini memodifikasi langsung hasil transformasi frekuensi sinyal. Contoh metode yang tergolong ke dalam teknik rana frekuensi adalah *spread spectrum*. Sinyal dalam rana spasial diubah kedalam rana frekuensi dengan menggunakan transformasi seperti *DCT* (*Discrete Cosine Transform*), *DFT* (*Discrete Fourier Transform*), dan *DWT* (*Discrete Wavelet Transform*).

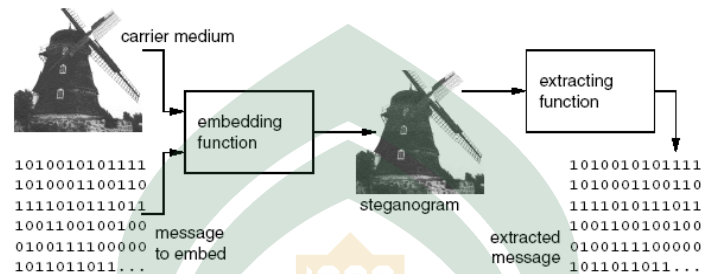
Pada tugas akhir ini akan dibahas mengenai teknik penyisipan pesan secara spasial karena metode Steganografi ini yang dibahas adalah metode LSB,

2. Metode *Least Significant Bit Insertion* (LSB)

pengubahan LSB (*Least Siqnificant Bit*) pada citra yang terkompresi sangat sulit diketahui secara kasat mata, sehingga metode ini sangat banyak digunakan. Metode ini memanfaatkan ketidak manpuan mata manusia dalam menemukan perbedaan antara gambar asli dengan yang sudah dimasukkan pesan.¹⁰ Pada Gambar II.3 ditunjukkan bahwa medium pembawa yang

¹⁰ Ibid.h.II.4.

disisipkan pesan dengan menggunakan suatu fungsi penyisipan, dalam hal ini *LSB* , menghasilkan *Stego-Image* yang tidak mengalami perubahan yang significant dari gambar aslinya.



Gambar II.1. Penyisipan pesan pada gambar

Untuk menjelaskan metode ini, digunakan citra digital sebagai *Image-Objek*. Setiap *Pixel* dalam citra digital berukuran 1sampai 3 byte . pada susunan bit didalam byte (1 byte = 8 bit), terdapat bit yang kurang berarti (*Least Significant Bitatau LSB*). Misalnya pada byte 00110011, maka bit *LSB*-nya adalah bit yang terletak paling kanan yaitu 1. Untuk melakukan penyisipan pesan, bit paling cocok untuk diganti denganbit pesan adalah bit *LSB*, sebab pengubahan bit tersebut hany akan mengubah nilai byte-nya menjadi satu lebih tinggi atau satu lebih rendah.

Sebagai contoh, urutan bit berikut ini menggambarkan 3 *Pixel* pada *Cover-Image* 24 bit.

(01010110	10111001	10000110)
(10001001	10001010	00010011)
(01011110	01111000	10101010)

Pesan yang akan disisipkan adalah karakter “**M**” , yang nilai binernya adalah **10010011**, maka yang akan dihasilkan *Stego-Image* dengan urutan bit sebagai berikut :

(01010111 10111000 10000110)

(10001001 10001010 00010010)

(01011111 01111001 10101010)

Perubahan yang tidak signifikan ini tidak akan tertangkap oleh indra manusia (jika media wadah adalah gambar, audio dan video).

Dalam contoh diatas penggantian *pixel* tak signifikan dilakukan secara terurut. Penggantian *pixel* tak signifikan juga dapat dilakukan secara tidak terurut, bahkan hal ini dapat meningkatkan tingkat keamanan data.

Pada gambar *Bitmap* 24-bit , tiap *pixel*-nya mengandung 24-bit kandungan warna atau 8 bit untuk masing-masing warna dasar (R, G dan B), dengan kisaran nilai kandungan antara 0(00000000) sampai 255(11111111) untuk setiap warna . perubahan *LSB* ini pada gambar jenis ini hanya akan merubah 1 nilai dari 256 nilai sehingga gambar hasil Steganografi akan sulit dibedakan dengan gambar aslinya.¹¹

Steganografi dengan metode *LSB* juga hanya mampu menyimpan informasi dengan ukuran yang sangat terbatas. Misalnya suatu citra-24 bit (R=8, G=8, B=8) digunakan sebagai Tempat untuk menyimpan data berukuran 100 bit, jika masing-masing komponen warnanya (RGB)

¹¹ Ibid.h.II.5.

digunakan satu pixel untuk menyimpan informasi rahasia tersebut, maka setiap pixelnya disimpan 3 bit informasi, sehingga setidaknya dibutuhkan citra wadah berukuran 34 pixel. Jadi suatu citra 24-bit jika digunakan untuk menyimpan informasi rahasia hanya mampu menampung informasi berukuran 1/8 dari ukuran citra penampung tersebut.¹²

Kapasitas gambar maksimum pesan yang dapat ditampung adalah panjang gambar x lebar gambar x 3 bit.¹³ Sebagai contoh, *Desktop* umum berukuran 1024 *pixel* x 768 *pixel*. Jadi ukuran pesan maksimum pada gambar dengan ukuran tersebut adalah 2.359.296 bit, atau sebanyak 294.912 karakter (1 karakter = 1 byet atau 8 bit). Selanjutnya jika *file* lebih besar dari *Image* maka akan memanfaatkan metode *Resize Image* dan *Kompres* pada *file*. untuk menutupi kelemahan yang dimiliki oleh Metode *Least Significant Bit*.

3. Citra Digital

Citra adalah gambar pada bidang dwimantra (dua dimensi). Secara matematis. Citra merupakan fungsi menerus dari intensitas cahaya pada bidang dwimantra.¹⁴

Terdapat dua jenis citra yaitu :

- a. Citra Diam yaitu citra tunggal yang tidak bergerak . contoh dari citra diam foto.

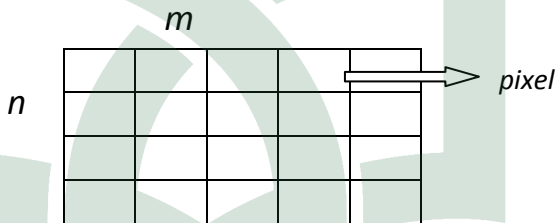
¹² Muhammad Hakim, *Op., cit.*h.1.

¹³ Westfeld, A., Pfitzmann, : *Attacks On Steganographic System*(1999).

¹⁴ Munir, Rinaldi. *Kriptografi*. Program Studi Teknik Informatika, (Bandung : Institut Teknologi Bandung, 2006).

- b. Citra bergerak yaitu rangkaian citra diam yang ditampilkan secara beruntun sehingga memberi kesan pada mata sebagai gambar yang bergerak. Contoh dari citra bergerak Video.¹⁵

Citra digital adalah suatu matriks yang terdiri dari baris dan kolom dimana setiap pasangan indeks baris dan kolom menyatakan suatu titik citra. Nilai matriksnya menyatakan tingkat kecerahan titik tersebut, yang disebut sebagai pixel.¹⁶ Citra digital sering kali dipresentasikan sebagai matriks $m \times n$, dimana element matriks adalah pixel. Contoh representasi matriks digambarkan pada Gambar II.2. sebagai berikut :



Gambar II.2. Representasi matriks citra.

4. Citra Bitmap

Bitmap adalah representasi dari citra grafis yang terdiri dari susunan titik yang tersimpan di memori komputer. Dikembangkan oleh *Microsoft* dan nilai setiap titik diawali oleh satu bit data untuk gambar hitam putih, atau lebih bagi gambar berwarna. Ukuran sebenarnya untuk n -bit (2^n warna) *bitmap* dalam byte dapat dihitung:

¹⁵ Ibid.

¹⁶ Gonzales, R.C., Woods, R.E. *Digital Image Processing*, (Addison-Wesley Publishing Company, 1992).

ukuran *file* BMP , $= 54 + 4.2^n + \frac{\text{Lebar} \cdot \text{tinggi} \cdot n}{8}$ dimana tinggi dan lebar dalam pixel.¹⁷

Kerapatan titik-titik tersebut dinamakan resolusi, yang menunjukkan seberapa tajam gambar ini ditampilkan, ditunjukkan dengan jumlah baris dan kolom, contohnya 1024x768. Untuk menampilkan citra *bitmap* pada monitor atau mencetaknya pada printer, komputer menterjemahkan *bitmap* ini menjadi pixel (pada layar) atau titik tinta (pada printer). Beberapa format *file bitmap* yang populer adalah BMP, PCX dan TIFF.

Citra *bitmap* sering disebut juga dengan citra raster. Citra *bitmap* menyimpan data kode cira secara digital dan lengkap (cara penyimpanannya adalah per *pixel*). Citra *bitmap* dipresentasikan dalam bentuk matriks atau dipetakan dengan menggunakan bilangan biner atau sistem bilangan lain. Citra ini memiliki kelebihan untuk memanipulasi warna, tetapi untuk mengubah objek lebih sulit. Tampilan *bitmap* mampu menunjukkan kehalusan gradasi bayangan dan warna dari sebuah gambar. Oleh karena itu, *bitmap* merupakan media elektronik untuk gambar-gambar dengan perpaduan gradasi warna yang rumit, seperti foto dan lukisan digital. Citra *bitmap* biasanya diperoleh dengan cara *scanner*, kamera digital, video *capture*, dan lain-lain.

Bitmap merupakan pemetaan bit. Gambar grafis komputer yang terdiri atas titik-titik yang membentuk baris dan kolom. Citra yang terbentuk terdiri

¹⁷ Munir, Rinaldi, *Op., cit.*

atas titik dan *pixel*. Seperti yang telah dijelaskan sebelumnya, pada citra raster, citra terdiri atas barisan *pixel* dan bukan vektor yang memiliki koordinat. Satu titik direpresentasikan oleh satu atau lebih bit data. Makin banyak bit yang digunakan untuk mempresentasikan satu titik, makin banyak warna dan bayangan abu-abu yang bisa digambarkan.

Citra BMP (*Bitmap*) adalah sebuah *Image* grafis yang disusun dari *pixel-pixel* dan dikonversi kedalam bits biasanya digunakan dalam *Microsoft Windows*. Gambar *Bitmap* adalah suatu gambar yang dipecah-pecah menjadi grid-grid (petak-petak).¹⁸

Citra digital *bitmap* atau yang disingkat BMP merupakan media yang akan digunakan untuk tempat menyembunyikan pesan pada tugas akhir ini. Format gambar BMP ini merupakan berkas yang memiliki ukuran yang cukup besar dan resolusi yang tinggi sehingga ukuran pesan yang dapat disisipkan semakin besar pula, disamping itu dengan resolusi tinggi yang dimiliki oleh BMP maka perubahan yang nampak pada gambar yang telah disisipkan pesan tidak terlalu signifikan. Format standar yang digunakan oleh Windows untuk menyimpan *file* gambar tersebut. BMP Mengandung lebih banyak informasi gambar dari JPEG. Saat ini, 24 bit per *pixel file* BMP adalah yang paling umum di semua 1, 4, 8, 15, 24, 32, 64 bit per *pixel file* BMP.

¹⁸ TIK, *Mengenal program Grafis*, (Yogyakarta :SMA Negeri 1 Yogyakarta. 2008), h1.

Kelebihan tipe *file* BMP adalah dapat dibuka oleh hampir semua program pengolah gambar. Baik *file* BMP yang terkompresi maupun tidak terkompresi, *file* BMP memiliki ukuran yang jauh lebih besar daripada tipe-tipe yang lain.

Operasi – operasi yang dilakukan di dalam pengolahan citra banyak ragamnya. Namun, secara umum operasi pengolahan citra dapat diklasifikasikan dalam beberapa jenis antara lain:

a. Perbaikan kualitas citra (*image enhancement*)

Jenis operasi ini bertujuan untuk memperbaiki kualitas citra dengan cara memanipulasi parameter – parameter citra. Dengan operasi ini, ciri – ciri khusus yang terdapat didalam citra lebih ditonjolkan.¹⁹ Contoh – contoh operasi perbaikan citra :

- 1) Perbaikan kontras gelap/terang
- 2) Perbaikan tepian objek (*edge enhancement*)
- 3) Penajaman (*sharpening*)
- 4) Pemberian warna semu (*pseudocoloring*)
- 5) Penapisan derau (*noise filtering*)

b. Pemugaran citra (*image restoration*)

Operasi ini bertujuan untuk meminimumkan/menghilangkan cacat pada citra. Tujuan pemugaran hampir sama dengan operasi perbaikan citra.²⁰

¹⁹ Rinaldi Munir, *op. Cit.*, h.9

²⁰ Ibid.

Bedanya pada pemugaran citra penyebab degradasi gambar diketahui.

Contoh – contoh pemugaran citra :

- 1) Penghilangan kesamaran (*deblurring*)
- 2) Penghilangan derau (*noise*)

c. Pemampatan citra (*Image compression*)

Jenis operasi ini dilakukan agar citra dapat direpresentasikan dalam bentuk yang lebih kompak sehingga memerlukan memori yang lebih sedikit.²¹ Hal penting yang harus diperhatikan dalam pemampatan adalah citra yang telah dimampatkan harus tetap mempunyai kualitas gambar yang bagus. Contoh metode pemampatan citra adalah metode JPEG.

d. Segmentasi citra (*image segmentation*)

Jenis operasi ini bertujuan untuk memecah suatu citra ke dalam beberapa segmen dengan suatu kriteria tertentu.²² Jenis operasi ini berkaitan erat dengan pengenalan pola. Proses segmentasi kadangkala diperlukan untuk melokalisasi objek yang diinginkan dari sekelilingnya.

e. Pengorakan citra (*image analysis*)

Jenis operasi ini bertujuan menghitung besaran kuantitatif dari citra untuk menghasilkan deskripsinya.²³ Teknik pengorakan citra mengekstraksi ciri – ciri tertentu yang membantu dalam identifikasi objek. Contoh – contoh operasi pengorakan citra :

²¹ Ibid, h.10

²² Ibid, h.11

²³ Ibid.

- 1) Pendeteksi tepi objek (*edge detection*)
 - 2) Ekstraksi batas (*boundary*)
 - 3) Representasi daerah (*Region*)
- f. Rekonstruksi citra (*image reconstuction*)

Jenis operasi ini bertujuan untuk membentuk ulang objek dari beberapa citra hasil proyeksi.²⁴ Operasi rekonstruksi citra banyak digunakan dalam bidang medis. Misalnya beberapa foto *rontgen* dengan sinar X digunakan membentuk ulang gambar organ tubuh.

Struktur berkas citra tipe bitmap 24-bit²⁵

Properti wadah (cover) yang digunakan dalam makalah ini adalah berupa citra bitmap 24-bit, struktur data citra tersebut adalah sebagai berikut:

BITMAPFILEHEADER
BITMAPINFOHEADER
RGBQUAD array
Color-index array

Gambar II.3. Struktur file bitmap

1) BITMAPFILEHEADER

Merupakan header dari *file*, berisi informasi tipe *file* ("BM"), ukuran *file* dalam byte, dan informasi jumlah offset byte antara header dan data bitmap yang sebenarnya.²⁶

²⁴ Ibid.

²⁵ Muhammad Hakim, *Op. Cit.*, h.4.

2) BITMAPINFOHEADER

Menyimpan informasi ukuran panjang dan lebar *file* dalam satuan pixel, format warna (jumlah bidang warna / bits-per-pixel), informasi apakah bitmap terkompresi atau tidak serta tipe kompresinya, jumlah data bitmap dalam byte, resolusi, dan jumlah warna yang digunakan.²⁷

3) RGBQUAD array

Berisi informasi intensitas RGB untuk setiap komponen warna pada pallete.²⁸

4) Color-index Array

Merupakan data *file* bitmap yang sebenarnya, pada bagian inilah informasi dapat disimpan dengan teknik steganogeafi metode LSB.²⁹

5. Compress Format ZLIB dengan Metode Lempel Ziv Welch (LZW)

Menurut David Salomon (2000), pada prinsipnya kompresi data dapat dicapai dengan mereduksi redundancy. Terdapat banyak metode untuk kompresi data yang sudah dikenal saat ini. Metode-metode ini berdasarkan pada ide-ide yang berbeda, dan cocok untuk tipe data yang berbeda pula, serta hasil yang berbeda. Meskipun demikian, semua metode

tersebut berdasarkan pada prinsip yang sama yakni semuanya memampatkan data dengan menghilangkan redundancy dari data asli dalam

²⁶ Ibid.h.4.

²⁷ Ibid.h.4.

²⁸ Ibid.

²⁹ Ibid

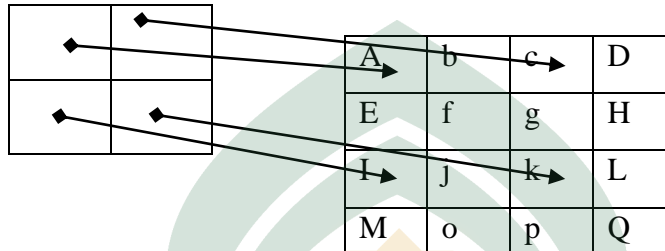
file sumber. Metode-metode kompresi data secara keseluruhan dapat dibagi ke dalam dua kelompok besar yakni kelompok metode kompresi data boleh-hilang (*lossy compression*) dan metode kompresi data tak-hilang (*lossless compression*) (Mark Nelson, 1996). Metode kompresi data boleh-hilang (*lossy*) yakni suatu metode kompresi data yang menghilangkan sebagian “informasi” data dari *file* asli (*file* yang akan dimampatkan) selama proses kompresi berlangsung dengan tidak menghilangkan (secara signifikan) informasi yang ada dalam *file* secara keseluruhan. Sedangkan algoritma kompresi data *lossless* adalah metode kompresi data yang mana tidak ada “informasi” data yang hilang atau berkurang jumlahnya selama proses kompresi. Sehingga setelah proses dekompresi jumlah bit (*byte*) data atau informasi dalam keseluruhan *file* hasil sama persis dengan *file* aslinya.³⁰

6. Ekspansi Wadah dengan Metode Resize

Dalam melakukan perbesaran ukuran citra, maka setiap pixel pada citra asal harus dipetakan pada sekumpulan pixel yang ada pada citra yang berukuran lebih besar. Tingkat efektivitas dari *algoritma* untuk menghasilkan perbesaran citra tersebut sangat menentukan citra hasil perbesaran. Pada umumnya *algoritma* yang mampu menghasilkan perbesaran citra dengan kualitas terbaik membutuhkan waktu pemrosesan dan sumber daya (*resource*) lain yang lebih besar pula.

³⁰ Hernawan Sulistyanto, *Kompresi Data Lossless dengan Metode Lempel-Zip* (Teknik Elektro Universitas Muhammadiyah Surakarta, 2003).h.33.

Misalkan dikehendaki perbesaran citra yang berukuran 2×2 dengan faktor perbesaran citra sebesar 2, maka hasil perbesaran citra adalah citra baru berukuran 4×4 .



Gambar II.4. Pembesaran citra 2×2 dengan faktor pembesar

Pada perbesaran citra pada Gambar II.4 diberikan ilustrasi perbesaran citra ukuran 2×2 dengan faktor perbesaran 2 sehingga menghasilkan citra baru berukuran 4×4 . Terlihat pada gambar tersebut setiap *pixel* dari citra asal (citra ukuran 2×2) dipetakan pada *pixel – pixel* citra hasil (a,c,i, dan k). Adapun *pixel-pixel* yang tersisa pada citra hasil juga harus diisi dengan nilai yang sesuai, sehingga tidak merusak hasil dari pemetaan ini.

Teknik termudah (tersederhana) dan tercepat dalam pengisian informasi setiap *pixel* yang masih kosong pada citra hasil perbesaran tersebut adalah dengan cara melakukan duplikasi citra asal. Sehingga jika cara ini yang dipilih, nilai *pixel b* diduplikasi (copy) dari *pixel a* sehingga nilai *pixel b* pada citra hasil akan memiliki nilai yang sama dengan nilai *pixel a*, nilai *pixel d* diduplikasi dari nilai *pixel c*, sedangkan nilai – nilai *pixel* pada baris kedua

yaitu *pixel e* sampai *h* diperoleh dengan cara menduplikasi nilai – nilai pada baris pertama (*pixel a* samapi *d*).

Algoritma yang lebih baik untuk melakukan pengisian nilai dari *pixel-pixel* kosong pada berkas hasil perbesaran adalah dengan cara melakukan interpolasi nilai *pixel-pixel* kosong tersebut dari nilai *pixel-pixel* yang ada di sampingnya (*nearest neighbors*). Pada dasarnya terdapat banyak *algoritma* yang mengimplementasikan teknik ini, akan tetapi kemampuan *algoritma* tersebut bervariasi. Tantangan terbesar dalam pengimplementasian *algoritma* ini terdapat pada kecepatan pemrosesan. Salah satu teknik yang cukup sederhana dengan menggunakan metode *Algoritma resize* yaitu dengan menghitung nilai rata-rata dari *pixel-pixel* yang terdapat pada sebelah kanan dan kiri *pixel* yang hendak diberi nilai. Dengan metode ini nilai-nilai pada *pixel-pixel b* dan *j* pada Gambar 4 diperoleh dengan menghitung masing – masing nilai rata – rata pada *pixel a* dan *c* serta *i* dan *k*. Nilai-nilai warna dari *pixel e* diperoleh dengan cara menghitung nilai rata – rata dari nilai warna pada *pixel a* dan *i* begitu juga untuk nilai -nilai warna pada *pixel f* dan *g*, yaitu nilai rata – rata dari *b* dan *j* serta *c* dan *k*. adapun cara penentuan nilai-nilai *pixel* yang tidak memiliki dua tetangga (kiri-kanan) yaitu *pixel-pixel* yang terletak di sudut gambar seperti *pixel d* ditentukan dengan cara melakukan duplikasi dari nilai yang ada pada tetangganya yang dalam hal ini adalah *pixel c*, sedangkan nilai *pixel h* diperoleh dari nilai rata-rata pada nilai

warna *pixel d* dan *l*. Proses yang dilakukan saat penggunaan algoritma ini adalah dengan cara:




- a. duplikasi nilai – nilai *pixel* citra asal pada citra hasil sesuai dengan tempatnya (misal: **a,c, i** , dan **k**)
- b. lakukan iterasi pada setiap baris tempat *pixel -pixel* pada langkah pertama diduplikasikan (misal: baris 1 dan 3) dan isi nilai-nilai *pixel* yang masih kosong dengan nilai baru yang diperoleh dengan menghitung nilai rata – rata dari *pixel-pixel* yang bertetanggan dengan *pixel* tersebut.
- c. lakukan iterasi pada baris – baris yang masih belum memiliki nilai dan isi nilai pada setiap *pixel*nya dengan cara menghitung nilai rata – rata dari nilai warna *pixel* pada bagian atas dan bawahnya.
- d. baris terakhir (baris ke-4) diisi dengan cara menyalin dari nilai yang ada pada baris ketiga.

Adapun salah satu cara penghitungan nilai rata-rata dari dua *pixel* adalah dengan menghitung nilai rata-rata dari komponen warna (red, green, blue) dari kedua *pixel* dan melakukan pencarian nilai yang paling mendekati dengan nilai tersebut pada *pallette*. Meskipun teknik penghitungan rata – rata ini cukup masuk akal akan tetapi pada dasarnya teknik ini bukanlah teknik yang paling baik dalam menghitung nilai rata-rata dari dua buah *pixel*.

7. Diagram Alir (Flowchart)

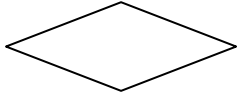
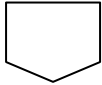



Didalam pemograman sangat dikenal dengan diagram alir (*flowchart*). Diagram alir (*flowchart*) digunakan untuk membantu analisis dan programmer untuk memecahkan masalah dalam pemograman. Diagram alir (*flowchart*) adalah gambaran secara grafik yang terdiri dari symbol-simbol dari algoritma-algoritma dalam suatu program, yang menyatakan arah dari alur program. *Flowchart* merupakan suatu teknik untuk menyusun rencana program telah diperkenalkan dan telah dipergunakan oleh kalangan programmer computer sebelum *algoritma* menjadi populer. *Flowchart* adalah untaian symbol gambar (*chart*) yang menunjukkan aliran (*flow*) dari proses terhadap data.³¹ *Flowchart* juga sering digunakan untuk memakai Diagram aliran suatu program. Symbol *flowchart* dapat diklasifikasikan menjadi symbol untuk program.

Table II.1. keterangan Gambar diagram alir Program³²

Nama	Gambar	Keterangan
Terminator		Untuk memulai atau selesai
Proses		Menyatakan proses terhadap data
Input output		Menerima input atau menampilkan output

³¹ Suarga, M. Sc., M. Math., Ph. D., Algoritma Pemograman (Makassar : 2004).h.6.

³² Ibid.

Seleksi		Memilih aliran berdasarkan syarat
Off-page Connector		Penghubung halaman yang berbeda
Alir		Aliran Program
Display/Monitoring		Untuk Menampilkan
Printer		Cetak Hasil Proses

8. Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah sebuah “bahasa” yang menjadi standar dalam industry untuk menentukan, visualisasi, merancang dan mendokumentasikan model dari sistem perangkat lunak. UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks.³³

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi perangkat lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun.

UML menyediakan beberapa notasi dan model standar yang digunakan sebagai alat komunikasi sebagai alat komunikasi bagi para pelaku

³³ Tesis Program Magister Manajemen.h.27

dalam proses analisis dan desain. Model dalam UML digunakan sebagai informasi dalam bentuk yang digunakan atau dihasilkan dalam proses pengembangan perangkat ³⁴

a. Notasi dalam UML

1) Actor



Gambar II.5. Notasi actor

Actor menggambarkan segala pengguna software aplikasi (user). Actor memberikan suatu gambaran jelas tentang apa yang harus dikerjakan software aplikasi. Sebagai contoh sebuah actor dapat memberikan input kedalam dan menerima informasi dari software aplikasi, perlu dicatat bahwa sebuah actor berinteraksi dengan use case, tetapi tidak memiliki kontrol atas use case. Sebuah actor mungkin seorang manusia, satu device, hardware atau sistem informasi lainnya.³⁵

2) Use Case



Gambar II.6. Notasi use case

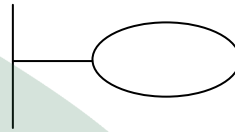
Use-case menjelaskan urutan kegiatan yang dilakukan *actor* dan sistem untuk mencapai suatu tujuan tertentu. Walaupun menjelaskan kegiatan, namun *use-case* hanya menjelaskan apa yang dilakukan oleh

³⁴ Ibid.h.36

³⁵ Ibid.

actor dan sistem bukan bagaimana *actor* dan sistem melakukan kegiatan tersebut.

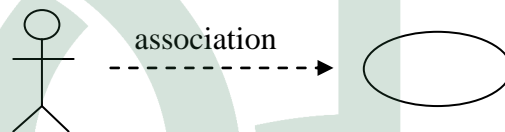
3) System Boundary



Gambar II.7. System boundary

Merupakan batas antara sistem dan actor, biasa dinotasikan dengan bujur sangkar sesuai dengan gambar diatas, semua *Use-case* harus berada didalam system boundary.³⁶

4) Relationship



Gambar II.8. Association relationship

Relationship adalah koneksi antara model element Interaksi antara actor dan *use-case* dalam *use-case* model biasanya digunakan *association relationship* yaitu :³⁷

<<uses>>

Hubungan *uses* menunjukkan bahwa prosedur dari *use-case* merupakan bagian dari prosedur yang menggunakan *use-case*. Tanda panah menunjukkan keadaan tidak mengakibatkan pemanggilan prosedur

³⁶ Willy Sudiarto Raharjo, Aditya Wikan Mahastama *Permodelan System Perangkat Lunak (Uses Case UML)* (Univ Kristen Duta Wacana, PSPL).

³⁷ Tesis Program Magister Manajemen., *Op. cit.*

dalam menggunakan *use-case*. Relasi *uses* antara *use-case* ditunjukkan dengan panah generalisasi dari *use-case*. *Use-case* yang dilakukan secara berulang, digunakan untuk meminimalkan pekerjaan

<<extend>>

Hubungan *extend* antar *use-case* berarti bahwa suatu *use-case* merupakan tambahan kegunaan dari *use-case* yang lain jika kondisi atau syarat tertentu dipenuhi. Jika prosedur dari *use-case* merupakan alternatif untuk menjelaskan *use-case* lain. *Use-case* akan dikerjakan apabila salah satu syarat terpenuhi.

<<include>>

Hubungan *include* menggambarkan suatu *use-case* seluruhnya meliputi kegunaan dari *use-case* lainnya. Sebuah *use-case* dapat meng-*include* fungsionalitas *use-case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use-case* yang di-*include* dieksekusi secara normal. Sebuah *use-case* dapat di-*include* oleh lebih dari *use-case* lain, sehingga duplikasi fungsional dapat dihindari.

BAB III

METODE PENELITIAN

A. Jenis Penelitian

Dalam melakukan penelitian ini, jenis penelitian yang digunakan adalah penelitian kuantitatif. Penelitian kuantitatif yang dilakukan metode penelitian eksperimental. Dengan melakukan eksperimen terhadap variabel-variabel (input) untuk menganalisis output yang dihasilkan. Penelitian Eksperimental merupakan bentuk penelitian dimana peneliti (eksperimenter) dengan sengaja menguji coba objek yang terdapat pada perangkat lunak yang dibangun, selanjutnya mengamati dan mencatat hasil ujicoba yang dilakukan, dan kemudian melihat hubungan diberikan dan reaksi yang muncul dari Proses.

B. Metode Pengumpulan Data

Metode Penelitian yang akan digunakan adalah *Field Research* dan *Library Research*. Dalam *Field Research* dilakukan dengan meneliti langsung ke Objeknya yang akan diteliti caranya dengan :

1. *Observasi* yaitu melakukan pengamatan secara langsung terhadap beberapa perangkat lunak untuk enkripsi Pesan rahasia.

Sedangkan *Library Research* (Penelitian Kepustakaan) yaitu dengan mencari referensi dari Buku perpustakaan dan juga dari Internet, yang berhubungan

dengan Objek yang diteliti, dimana referensi tersebut dijadikan landasan teori dalam penyelesaian tugas akhir ini.

C. Alat dan Bahan

Perangkat lunak yang dibuat dikembangkan pada perangkat keras *notebook* Asus A42F, dengan spesifikasi:

1. Processor P6200 2.13Ghz
2. Hardisk 320 GB
3. Memoiri RAM 2 GB

Sedangkan spesifikasi perangkat lunak yang dipakai, sebagai berikut:

1. Sistem operasi Microsoft Windows XP2
2. Delphi 7
3. ZlibEx (Zlib)

D. Lokasi Penelitian Implementasi

Lokasi Penelitian implementasi perangkat lunak ini guna memanfaatkan perangkat lunak ini untuk masyarakat adalah dengan memanfaatkan dalam dunia pendidikan yaitu Dinas pendidikan.

E. Jadwal Penelitian

Adapaun waktu yang diperlukan dalam penyusunan tugas akhir ini yaitu :

No	Kegiatan	Februari				Maret				April				Mei				Juni				Juli							
		Minggu																											
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4				
1	Pengumpulan Data																												
2	Analisis Sistem																												
3	Pembuatan Program																												
4	Pengujin Sistem																												

Gambar III.1 Jadwal penelitian

BAB IV

ANALISIS DAN PERANCANGAN PERANGKAT LUNAK

Perangkat lunak yang akan dibuat dalam tugas akhir akan dibangun pada perangkat lunak dan dinamakan StegoBitmap. Fungsi utama dari perangkat lunak StegoBitmap ini adalah menyisipkan sebuah pesan ke dalam *Image* dan mengekstraknya kembali, dimana pada saat menjalankan fungsi penyisipan pesan maka perangkat lunak akan menerima masukan berupa *Image* dalam format *Bitmap*, melalui proses kompres dan Ekspansi wada, kemudian pesan rahasia . Keluaran atau *output* dari perangkat lunak ini berupa *Image* berformat *Bitmap* yang mengandung sebuah pesan rahasia.

Pada fungsi ekstraksi pesan, perangkat lunak akan menerima masukan berupa *Image* yang berisi pesan dan kemudian dilakukan proses Decompres untuk mengembalikan kapasitas semula. *Output* yang dihasilkan dari fungsi ini adalah pesan yang tersembunyi di dalam gambar.

A. Analisis Perangkat Lunak

Perangkat Lunak yang akan dibangun pada tugas akhir ini adalah StegoBitmap dan dibangun pada lingkungan perangkat komputer pribadi, berikut ini akan dijelaskan Deskripsi umum Perangkat lunak, Alur perangkat lunak, kebutuhan perangkat lunak, serta analisis kelas.

1. Dekripsi Umum Perangkat Lunak

Perangkat lunak yang dibangun memiliki beberapa fungsi utama yaitu fungsi penyisipan pesan dan ekstrak pesan. Fungsi pendukungnya yaitu *resize* dan kompres *file* serta Dekompres.

Pada fungsi penyisipan pesan ini menggunakan metode *Least Significant Bit*. Perangkat lunak menerima masukan berupa citra *Bitmap* dan pesan. Pesan yang telah dikompres maupun yang tidak. Keluaran dari fungsi penyisipan pesan ini adalah citra *Bitmap* yang telah disisipi pesan didalamnya.

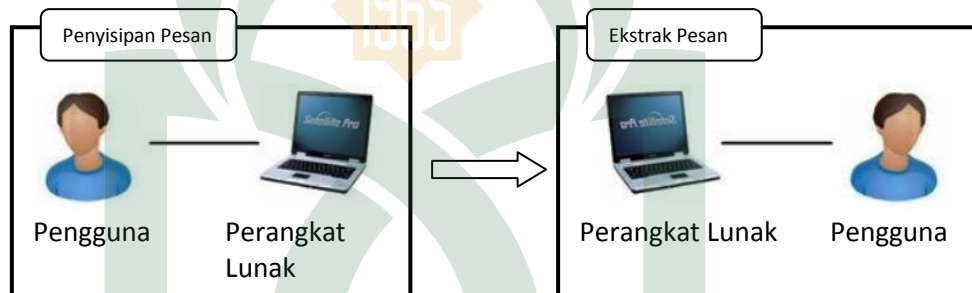
Fungsi Ekstrak pesan yaitu melakukan ekstrak pesan dari citra *Bitmap* yang sebelumnya telah mengalami proses penyisipan pesan menggunakan metode *Least Significant Bit*. Pada fungsi ekstrak pesan, perangkat lunak menerima masukan berupa citra *Bitmap* yang memiliki pesan yang digunakan pada fungsi penyisipan pesan. Keluaran dari fungsi ekstrak pesan ini adalah pesan yang disisipkan pada citra *Bitmap*.

Fungsi Kompres yaitu mengurangi kapasitas pada *file* yang akan disisipkan pada citra *Bitmap*. Sehingga proses penyisipan lebih cepat. Keluaran dari proses kompres ini adalah kapasitas *file* lebih kecil dari pada sebelumnya dan memiliki ekstensi *file* *zlib*.

Fungsi Ekspansi wadah yaitu memperbesar *pixel* gambar citra *Bitmap*. jika citra yang ingin disisipkan pesan tersebut lebih besar pesannya dari pada

citranya. Keluaran dari proses resize ini adalah *pixel* pada citra *Bitmap* lebih besar dari sebelumnya.

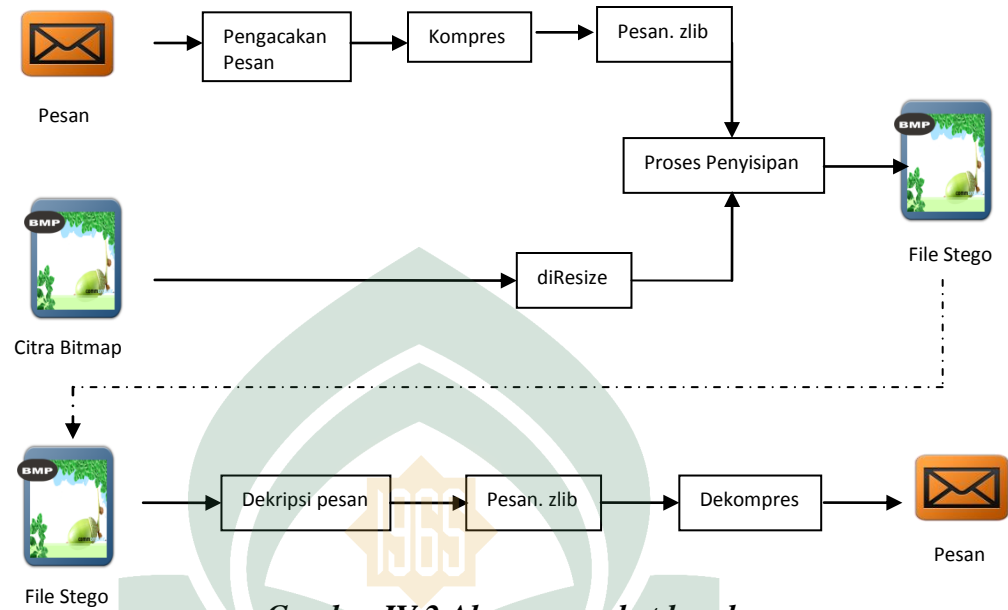
Fungsi Dekompres yaitu mengembalikan *file* yang hanya sudah di kompres Dan mengembalikannya menjadi kapasitas *file* aslinya. Sehingga proses ini tidak mengurangi dan menghilangkan isi pesan. Keluaran dari proses Dekompres ini adalah mengembalikan *file* yang *Zlib* menjadi *file* aslinya.



Gambar IV.1 Gambaran umum sistem

2. Alur Perangkat Lunak

Alur dari perangkat lunak dapat dilihat pada Gambar IV.2. Terlihat pada StegoBitmap mempunyai dua buah model utama yang memprementasikan dua buah fungsi utama yang sudah disebutkan pada subbab sebelumnya, yaitu model penyisipan pesan dan modul ekstraksi pesan.



Gambar IV.2 Alur perangkat lunak

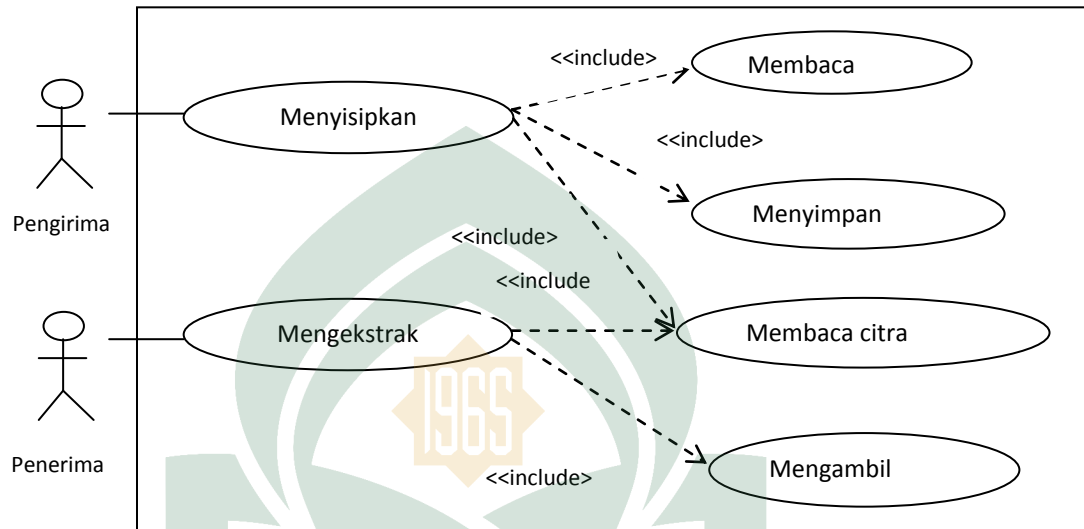
Modul penyisipan pesan memiliki masukan berupa pesan, citra dan *Bitmap*. Pesan yang ingin dimasukkan akan dikompres, dan kemudian akan dilakukan proses penyisipan pesan dan akan menghasilkan *file Stego Image*. Untuk modul Dekripsi pesan memiliki masukan berupa *Stego Image* yang diekstrak kemudian dilakukan dekompres untuk mengembalikan kapasitas aslinya dan kemudian menghasilkan pesan aslinya.

3. Use case Perangkat Lunak

Perangkat lunak yang dibangun pada tugas akhir ini diharapkan dapat melakukan hal-hal sebagai berikut :

- Menyisipkan pesan pada citra *Bitmap*
- Melakukan Kompres dan Dekompres
- Dapat melakukan Pengungkapan pesan yang disisipkan pada citra *Bitmap*

Untuk menggambarkan kebutuhan Perangkat lunak secara visual digunakan sebuah diagram *use-case* :³⁸



Gambar IV.3. Diagram use-case

Perangkat lunak memiliki enam buah *Use-case*. Dua buah *use-case* utama, penyisipan pesan dan pengungkapan pesan dan Terdapat dua person yaitu Penerima dan pengirim. *Use-case* pembacaan pesan digunakan untuk membaca pesan sedangkan *use-case* menyimpan pesan digunakan untuk menyimpan pesan di dalam suatu citra *Bitmap* menggunakan metode *Least Significant Bit*. Sedangkan *uses-case* mengambil pesan digunakan untuk mengambil pesan dari citra *Bitmap* yang telah mengalami proses penyisipan menggunakan metode *Least significant Bit*. *Use-case* membaca citra *Bitmap* digunakan untuk membaca citra *Bitmap* yang dimasukkan oleh pengguna, baik dalam proses penyisipan maupun pengungkapan pesan. penjelasan dapat dilihat pada table berikut :

³⁸ Willy Sudiarto Raharjo, *op.cit*.

Tabel IV.1 Use-case

NO.	Elemen Use- Case	KETERANGAN
1	Nama	<i>Menyisipkan Pesan</i>
	Deskripsi	Melakukan penyisipan pesan pada gambar
	Prekondis	Aplikasi atau sistem menampilkan menu utama
	Proses	Pengirim memilih menu menampilkan dialog proses penyisipan pesan
	Kondisi Akhir	Gambar telah berisi pesan
2	Nama	<i>Pengungkapan Pesan</i>
	Deskripsi	Melakukan Pengungkapan pesan ke dalam gambar
	Prekondis	sistem menampilkan menu utama dari perangkat lunak
	Proses	Pengguna memiliki menu untuk menampilkan dialog proses Pengungkapan pesan
	Kondisi Akhir	Pesan sudah di Dekripsi
3	Nama	<i>Membaca Pesan</i>
	Deskripsi	Melakukan pembacaan pesan berupa document
	Prekondis	<i>Use case</i> Melakukan Menyisipkan Pesan
	Proses	Sistem menampilkan <i>Dialog</i> proses Pengirim memasukkan pesan dan gambar Sistem menyimpan pesan yang dimasukkan
	Kondisi Akhir	Pesan sudah tersimpan
4	Nama	<i>Membaca Gambar</i>
	Deskripsi	Melakukan pembacaan citra <i>Bitmap</i>
	Prekondis	<i>Use-case</i> Melakukan Menyisipkan Pesan
	Proses	Sistem menampilkan <i>Dialog</i> proses Pengguna memasukkan gambar berformat <i>Bitmap</i> dan pesan Sistem melakukan validasi ukuran gambar Sistem menampilkan <i>Dialog</i> untuk menyimpan hasil

	Kondisi Akhir	Alamat gambar tersimpan
5	Nama	<i>Menyimpan Pesan</i>
	Deskripsi	Melakukan Penyimpangan pesan pada gambar
	Prekondisi	<i>Use-case</i> membaca pesan dilakukan
	Proses	Penerima memilih menu ekstrak pesan
	Kondisi Akhir	Pengirim menekan tombol “Proses Penyisipan” Sistem menampilkan Proses <i>Dialog</i> Sistem melakukan pembacaan pesan dan Gambar Sistem mengolah data tersebut dan menjadikan Gambar
6	Nama	<i>Mengambil Pesan</i>
	Deskripsi	Mengambil kembali pesan yang tersembunyi pada gambar
	Prekondisi	<i>Use-case</i> Membaca gambar sudah dilakukan dan mengungkap pesan
	Proses	Penerima menekan tombol “Proses pengungkapan” Sistem menampilkan Proses <i>Dialog</i> Sistem melakukan pembacaan data gambar Sistem mengolah data tersebut dan menjadikan pesan
	Kondisi Akhir	Pesan dihasilkan oleh system

B. Perancangan Sistem

1. Kebutuhan sistem

Perancangan system yang diperlukan meliputi :

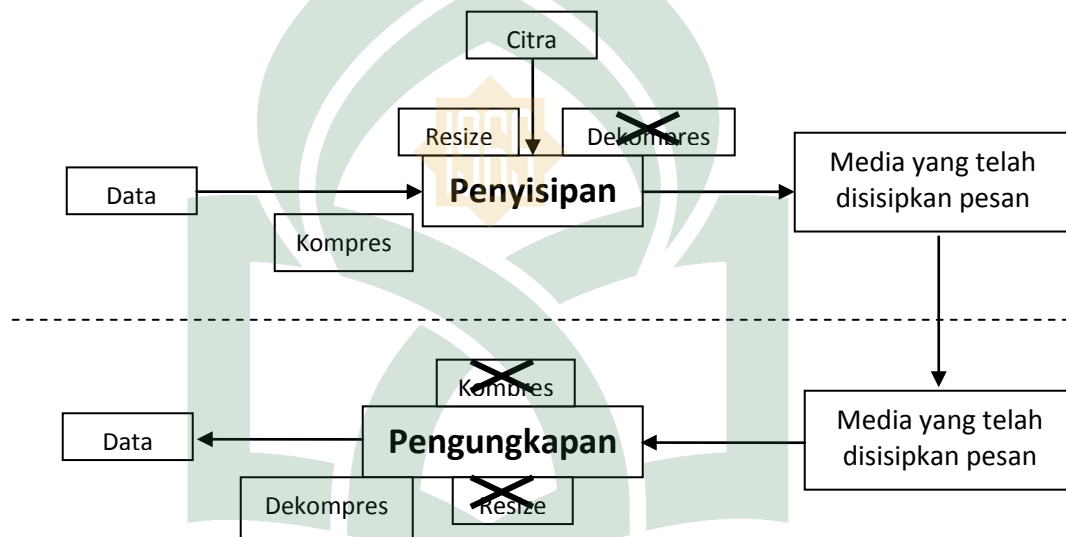
- Proses penyisipan data rahasia
- Proses pengungkapan data Rahasia kembali

Sistem komputerisasi akan mampu memenuhi kebutuhan tersebut dengan menggunakan borlan Delphi 7 sebagai bahasa pemograman makan akan

menghasilkan satu perangkat lunak Steganografi metode *Least Significant Bit* dengan teknik compress dan Ekspansi wadah.

2. Perancangan Program

Realisasi tahap Perancangan perangkat lunak. Proses penyisipan data rahasia dan proses pengungkapannya sebagai garis besar alurnya sebagai berikut.

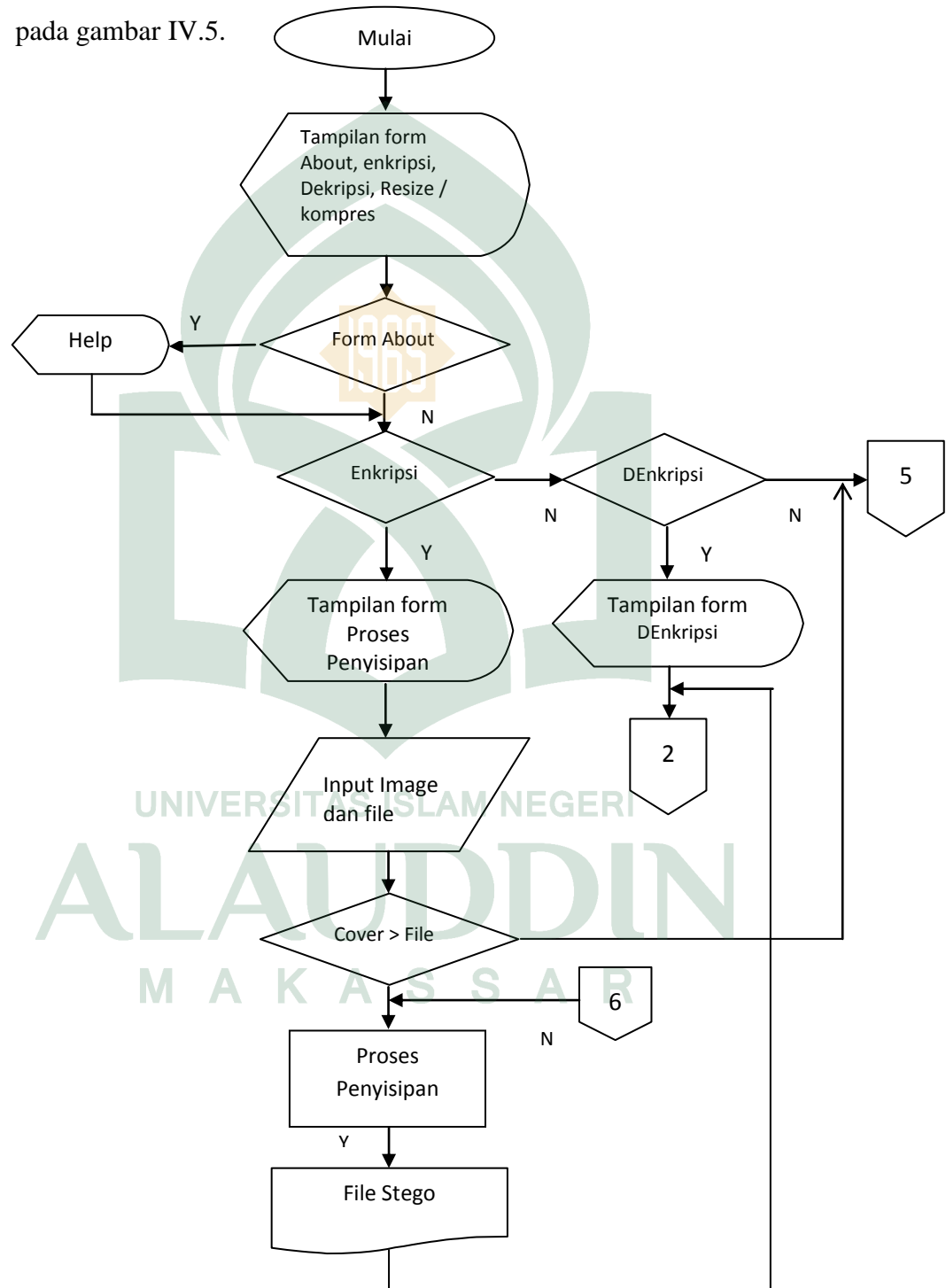


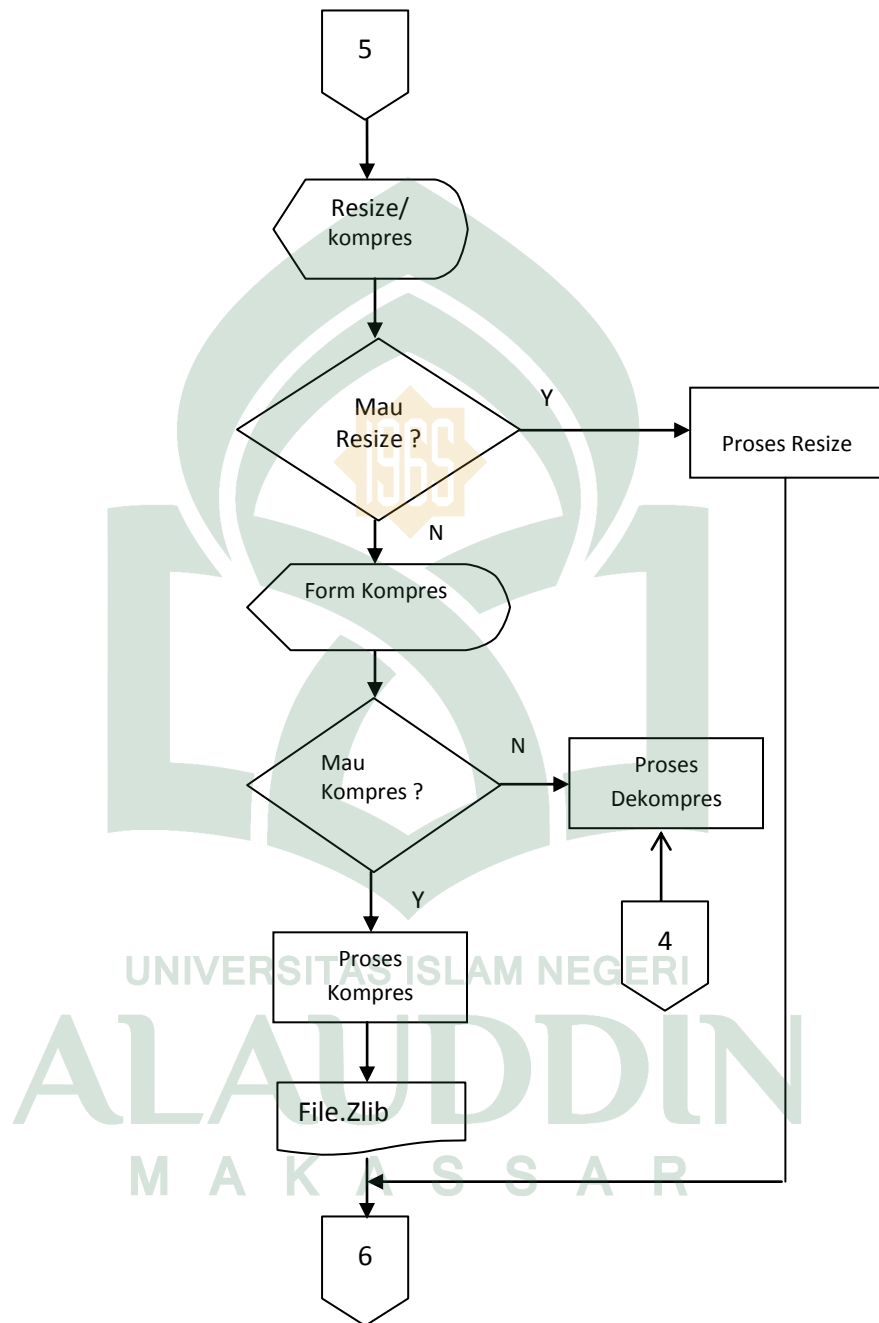
Gambar IV.4. Proses penyisipan dan pengungkapan data

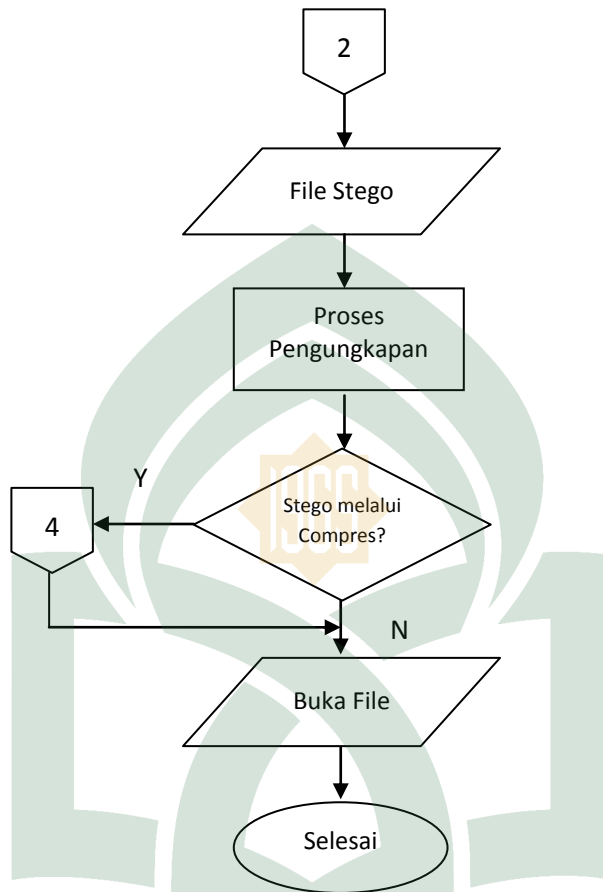
3. Perancangan Diagram Alir Perangkat Lunak

Program yang dijalankan akan menampilkan *form* pembuka yang akan tampil beberapa saat lalu menutup dan membuka *form* menu. *form* tersebut terdiri atas beberapa pilihan Menu yaitu Menu About berisikan petunjuk, Menu Enkripsi berisikan tombol untuk proses penyisipan, menu Deskripsi yang berisi Proses pengungkapan, Menu Resize/Compres berisikan Kompres *file* dan resize Gambar yang berfungsi untuk mengkompress *file* agar

memiliki kapasitas lebih kecil sedangkan untuk tombol *resize* berfungsi untuk memperbesar gambar. Diagram alirnya untuk penyisipan pesan, ditunjukkan pada gambar IV.5.







Gambar IV.5. Diagram alir perangkat lunak : Proses penyisipan proses pengungkapan, proses kompres dan resize

Berikut ini adalah penjelasan Diagram alir (flowchart) dari proses kerja perangkat lunak :

1. Dimulai dari tampilan awal yaitu atau ada menu about, enkripsi, Dekripsi dan resize/kompres.
2. Menu about ini berisikan tombol help yaitu petunjuk untuk menggunakan aplikasi StegoBitmap.

3. Setelah itu ke Menu berikutnya yaitu enkripsi. yang berisi pilihan untuk proses Penyisipan. Dengan cara, Data yang dimasukkan yaitu BMP (media image yang akan digunakan untuk menyembunyikan data), dan *file* sisip (data yang akan disisipkan dalam BMP), setelah itu dilakukan proses penyisipan. Dalam proses ini terjadi dua option yaitu:
 - a. Jika terjadi pesan error *file* atau *file* yang dimasukkan lebih besar dari kapasitas Image maka akan dilakukan proses resize atau kompres yang ada pada Menu resize/kompres. Proses tersebut bisa dilihat pada Gambar IV.5. dari proses resize hasilnya adalah 2x dari besar gambar sebelumnya sedangkan proses kompres mengurangi ukuran *file* atau size *file*, hasil dari proses kompres ini adalah menghasilkan *file* ekstensi Zlib.
 - b. tapi kalo *file* lebih kecil dari Image maka proses berlanjut ke proses Penyisipan pesan.
4. Selanjutnya proses penyisipan dilakukan. Proses tersebut bisa dilihat pada gambar IV.5, hasil dari Proses ini adalah *file Image* ekstensi BMP.
5. Menu Deskripsi ini berisikan pilihan untuk melakukan proses pengungkapan *file* gambar yang berekstensi BMP yang berisikan *file*. Proses tersebut bisa dilihat pada Gambar IV.5. hasil dari proses ini adalah *file* yang tidak berekstensi.

6. Untuk proses membuka *file* yang sudah melalui proses pengungkapan ini ada 2 option yaitu :

- a. Jika *file* tadi melalui proses Kompres maka *file* tersebut diubah menjadi *file* yang berekstensi *zlib*, bisa dilihat Gambar IV.6. berikut:

Dan kemudian lakukan Dekompres sehingga *file* yang disembunyikan tidak rusak dan informasinya tidak hilang, fungsi dari Dekompres ini adalah untuk mengembalikan kapasitas *file* aslinya sebelum dilakukan kompres.



Gambar IV.6. Proses ubah ekstensi file ke zlib

7. Kemudian *open with* sesuai ekstensi *file* yang sebelumnya disembunyikan.

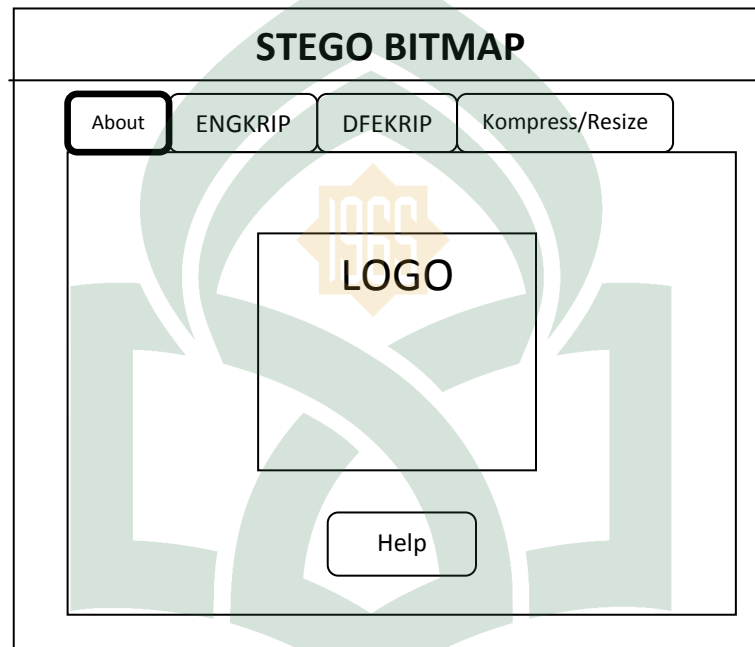
C. Perancangan interface

Berikut perancangan *interface* yang digunakan pada implementasi tugas Akhir ini :

1. *Interface* Menu Beranda (Utama)

Menu beranda adalah menu awal yang ditampilkan adalah menu About saat aplikasi di jalankan. Pada menu ini ditampilkan satu proses yaitu tombol Help petunjuk untuk menggunakan aplikasi ini. Sedangkan menu yang lainnya adalah Enkripsi (Proses penyisipan), Dekripsi (Proses Pengungkapan),

Resize/Kompres (proses *Resize* dan *Kompres*). jika Menu enkripsi dipilih maka akan menampilkan proses penyisipan dan jika tombol Dekripsi dipilih maka akan menampilkan proses Pengungkapan. Begitu juga dengan tombol *resize/kompres* akan menampilkan prosesnya.

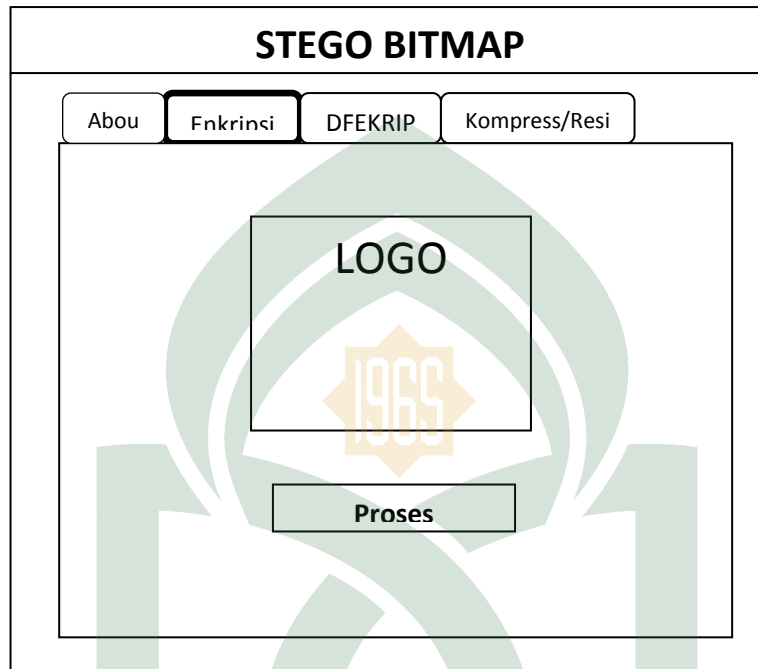


Gambar IV.7. Menu utama

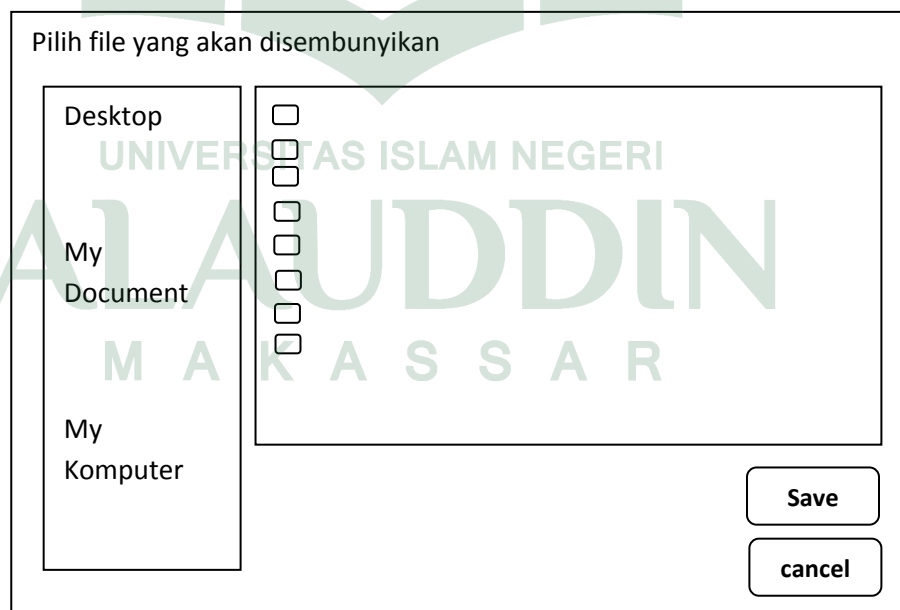
2. *Interface* Menu Enkripsi

Menu Enkripsi ini berisi tombol proses penyisipan pesan dan ditampilkan pada saat Menu Enkripsi dipilih pada menu berand. Pada saat tombol tersebut ditekan maka akan menampilkan yaitu *OpenDialog* dan *SaveDialog* . Yang pertama *OpenDialog* untuk memilih *file* yang akan di sembunyikan. Yang kedua muncul *OpenDialog* untuk memilih *Image*

sebagai yang akan disisipi *file*. ketiga akan muncul *SaveDialog* untuk menyimpan gambar dari hasil penyisipan. dilihat pada gambar berikut.



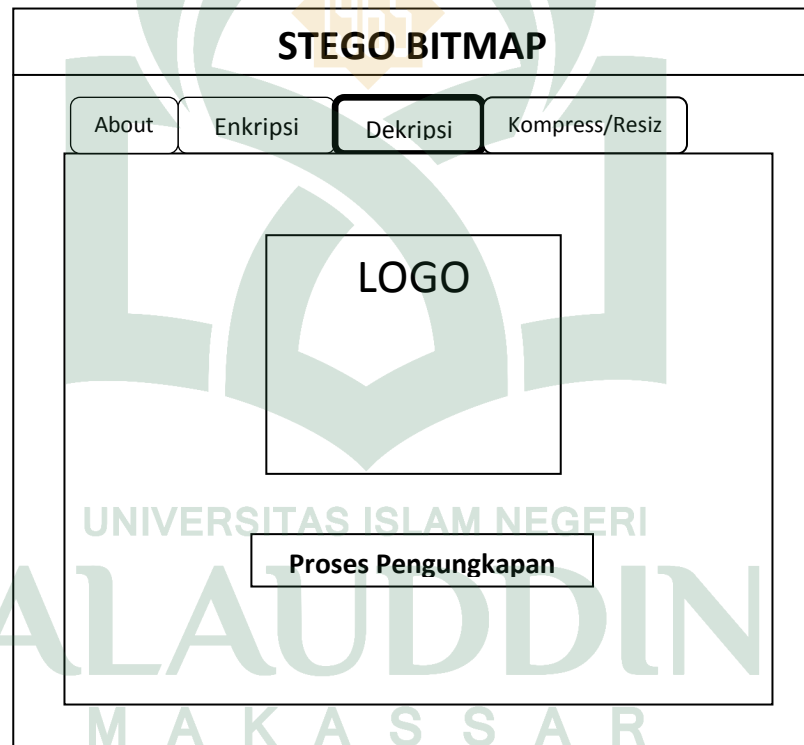
Gambar IV.8. Interface menu enkripsi pesan



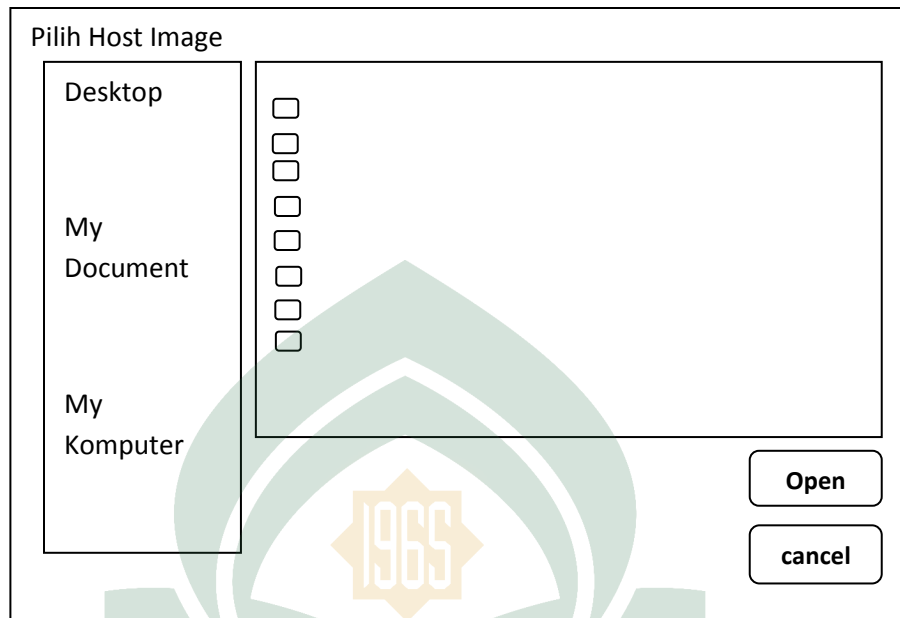
Gambar IV.9. Dialog untuk proses Penyisipan

3. *Interface* Menu Dekripsi

Proses Pengungkapan Pesan akan ditampilkan saat Menu Dekripsi dipilih pada menu awal. Pada proses Dekripsi ini akan ditampilkan satu tombol Proses Pengungkapan pesan dan pada saat tombol proses dipilih maka akan muncul *Dialog* untuk proses tersebut. Yang pertama *Dialog* untuk memilih kembali gambar yang telah disisipkan *file/data* sebelumnya. yang kedua *Dialog* untuk menyimpan *file* yang diekstrak tadi di tempat yang diinginkan.



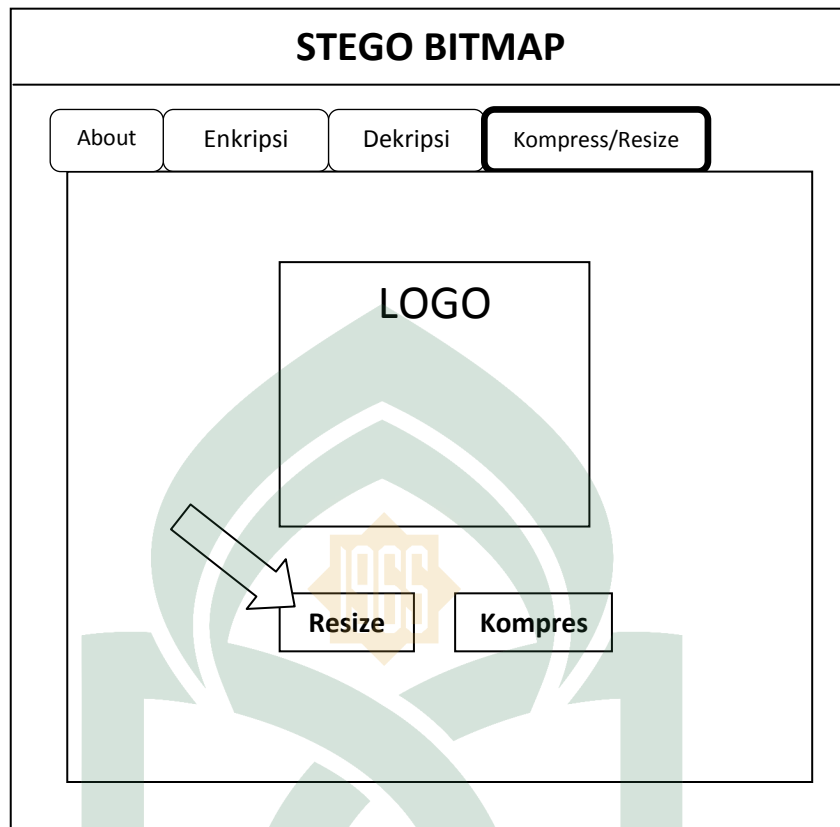
Gambar IV.10. Interface menu dekripsi pesan



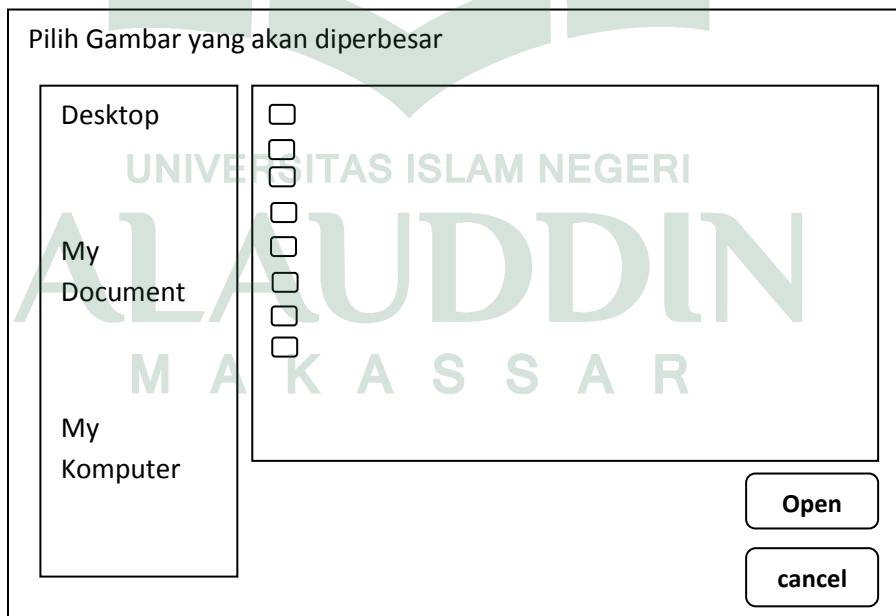
Gambar IV.11. Dialog pengungkapan pesan

4. Interface Menu Kompres dan Resize

Proses *Resize* dan Kompres akan ditampilkan saat tombol *Resize* dan Kompres dipilih pada menu awal. Pada proses tombol *resize* dan kompres ini akan ditampilkan prosesnya . untuk proses *Resize* akan menampilkan *Dialog*. *Dialog* untuk memilih gambar yang akan diperbesar. *Dialog* untuk menyimpan gambar yang diperbesar tadi di tempat yang diinginkan. Sedangkan untuk Proses Kompres akan ditampilkan *Form* untuk kompres dan dekompres *File*.



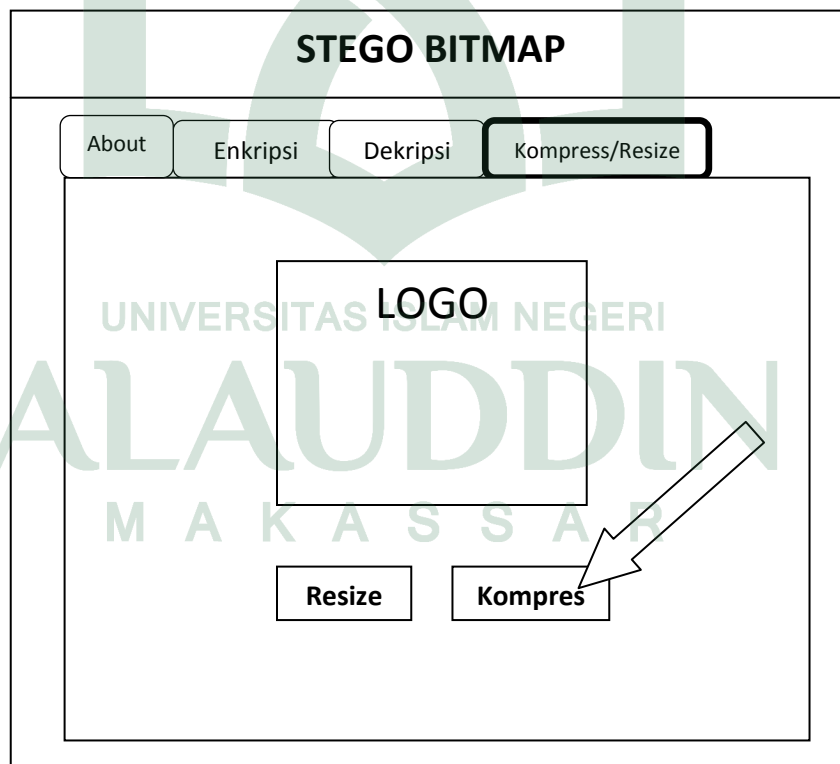
Gambar IV.12. Menu proses resize



Gambar IV.13. Dialog proses rezise

5. *Interface Menu Compres File/Data*

Proses *Compres File/Data* akan ditampilkan saat tombol *Compres* dipilih pada menu awal. Pada proses *Compres* ini akan ditampilkan Menu kompres dan Dekompres. proses kompres dengan cara memasukkan *file* yang akan di kompres klik kanan mouse pilih *add file* memilih *file* yang akan di kompres, pilih tombol kompres, akan muncul *Browser for folder* yaitu memilih tempat *file* hasil kompres, *DeCompres* untuk mengembalikan *file* yang telah di *Compres*, dengan cara pilih *add file* pilih *file* hasil kompresnya selanjutnya pilih tombol *Decompres*, sehingga akan muncul *Browser for folder* yaitu memilih tempat *file* hasil *Decompres*.



Gambar IV.14. Menu kompres

Form Kompres

<div style="display: flex; justify-content: space-around;"> Compres Dekompres </div>			
File	size	Compressed size	Ratio
<div style="border: 1px solid black; padding: 5px; display: inline-block;">Compres</div>			

Gambar IV.15. Menu proses kompres file

Form Kompres

<div style="display: flex; justify-content: space-around;"> Compres Dekompres </div>			
File	size	Compressed size	Ratio
<div style="border: 1px solid black; padding: 5px; display: inline-block;">Dekompres</div>			

Gambar IV.16. Menu proses dekompres file

BAB V

IMPLEMENTASI DAN PENGUJIAN SISTEM

A. Implementasi Perangkat Lunak Steganografi

Penjelasan implementasi perangkat lunak ini meliputi pembahasan dan Implementasi *interface* dari perangkat lunak steganografi.

1. Implementasi kelas

Setiap kelas pada perangkat lunak diimplementasikan dalam bahasa pemrograman Delphi. Sebuah kelas akan diimplementasikan dalam sebuah *file* dimana semua kelas tersebut sesuai dengan kelas perancangan yang telah diuraikan pada subbab sebelumnya.

Pada implementasi penyisipan, ukuran pesan yang ingin disisipkan oleh pengguna akan melewati proses pengujian validasi ukuran terlebih dahulu. Apabila ukuran dari pesan melebihi batas maksimum ukuran pesan yang akan disisipkan pada citra *Bitmap*, proses maka akan dilakukan proses Compress *file* dan *Resize Image* setelah itu akan dilakukan penyisipan pesan. Kelas-kelas yang telah dirancang akan diimplementasikan pada table berikut akan dilihat daftar implementasi kelas-kelas yang ada pada *stegoBitmap* serta keterangannya

Table V.1 Daftar Tabel Kelas Perancangan dan Implementasi

Nama kelas	Nama File	Keterangan
About/ Interface	Unit1.dpr	Kelas ini merupakan <i>form</i> yang akan menjadi jendela utama dari perangkat lunak. Kelas ini akan memanggil procedure StegoBitmap yang ada pada kelas lainnya.
Prosess	Unit2.dpr	Kelas ini berfungsi untuk melakukan proses Penyisipan dan ekstrak <i>file</i> atau data.
Resize	Unit3.dpr	Kelas ini mengimplementasikan resize gambar pada saat gambar ingin diperbesar pixelnya
Compress	Unit4.dpr	Kelas ini merupakan kelas yang akan melakukan proses kompres
	Zlibx	Adalah suatu aplikasi pendukung dari Delphi untuk melakukan compress dan Decompress data.

2. Implementasi *interface*

Implementasi *interface* dari perangkat lunak StegoBitmap sesuai dengan perancangan *interface* yang telah diuraikan pada bab IV. Dengan tambahan layar *error*. layar *error* akan menampilkan pesan kesalahan yang terjadi, dan hanya memiliki sebuah tombol “OK” untuk kembali. layar *error* ini akan muncul apabila salah satu dari hal berikut:

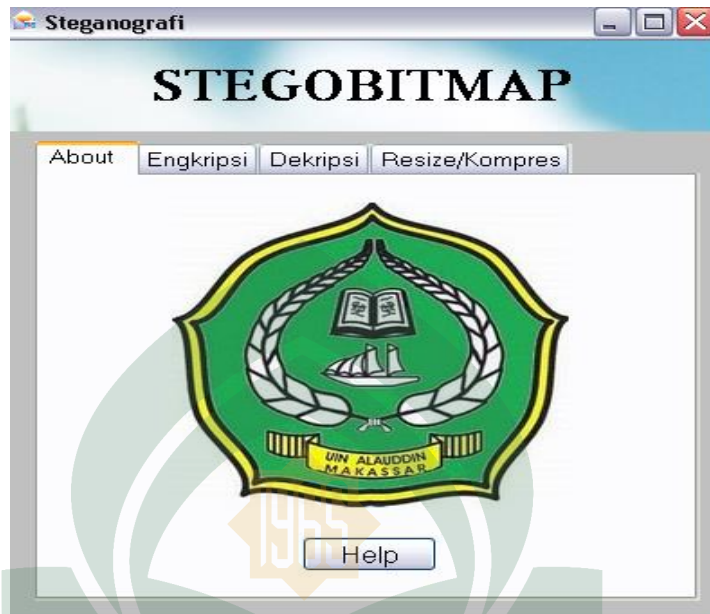
- a. Ukuran pesan yang disisipkan lebih besar dari pada kapasitas pesan yang dapat ditampung oleh citra *Bitmap* yang menjadi media penyisipan pesan.
- b. Citra *Bitmap* tidak memiliki kualitas 24-bit sehingga tidak memenuhi syarat untuk dijadikan media penyisipan pesan.

Sesuai dengan perancangan pada sub Bab 4 terdapat beberapa buah menu utama pada Perangkat Lunak StegoBitmap yaitu Menu About (*help*), Menu Enkripsi (proses penyisipan), Dekripsi (proses Pengungkapan), Menu kompres/resize (*Resize Image* dan Kompres data). Penjelasan mengenai implementasi Menu secara detail akan dijelaskan sebagai berikut :

- a. Menu Beranda About

Menu Beranda adalah menu utama dari aplikasi. Aplikasi akan menampilkan menu ini pertama kali saat aplikasi dijalankan. *Interface* pada Aplikasi dapat dilihat pada Gambar. V.1 berikut :

UNIVERSITAS ISLAM NEGERI
ALAUDDIN
M A K A S S A R



Gambar V.1 Menu beranda

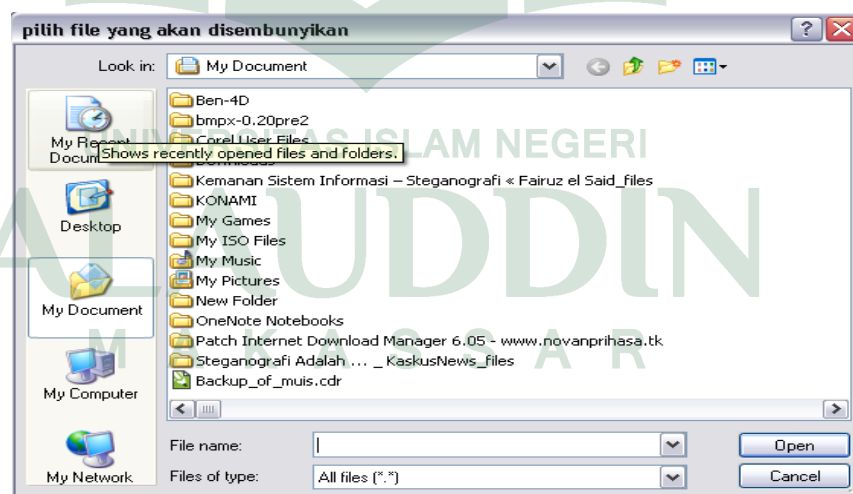
b. Menu Enkripsi (Proses penyisipan Pesan)

Menu proses penyisipan pesan ditampilkan pada saat pengguna memilih tombol “*Enkripsi*”. Pada Menu ini akan hanya ditampilkan tombol Proses Penyisipan, pada saat pilih tombol ini maka akan muncul *Dialog*. Memilih pesan yang akan disisipkan pada citra *Bitmap* yang dijadikan media penyisipan dan menyimpan hasil dari proses penyisipan. Implementasi proses penyisipan pesan untuk proses dapat dilihat pada Gambar V.2.sebagai berikut :



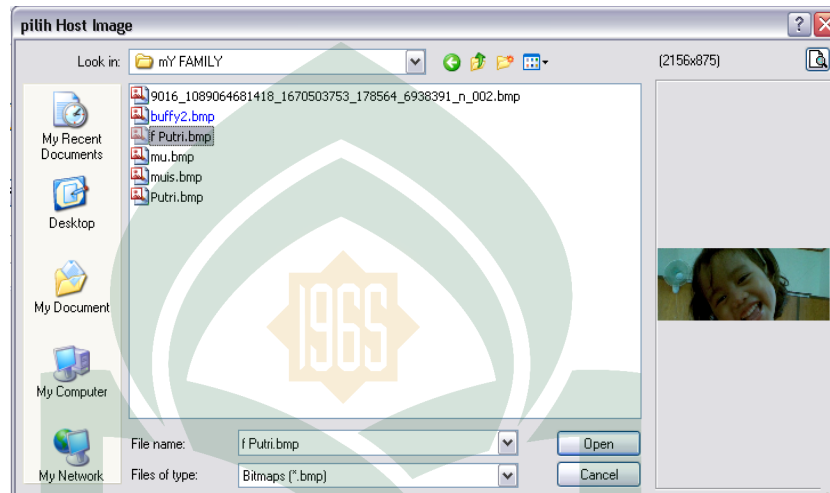
Gambar V.2 Menu proses enkripsi pesan

Pada Gambar V.3. ini muncul *Dialog* yaitu memilih *file* yang akan disisipkan sebuah pesan dibawah ini :



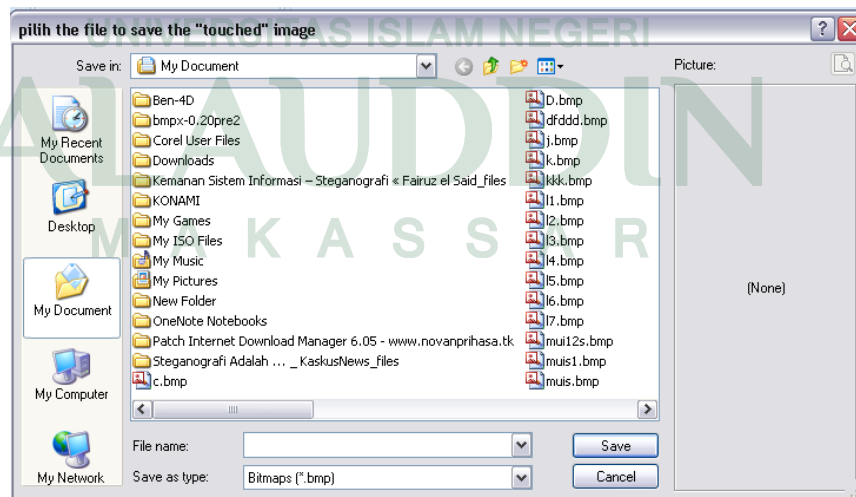
Gambar V.3 Dialog memilih file

Kemudian akan muncul *Dialog* untuk memilih gambar yang akan disisipkan pesan, Implementasinya dapat dilihat pada Gambar V.4. berikut:



Gambar V.4 Dialog memilih image

Selanjutnya akan tampil *Dialog* untuk memilih lokasi hasil dari proses penyisipan pesan. Implementasinya dapat dilihat pada Gambar V.5. berikut:



Gambar V.5 Menyimpan gambar yang telah disisipi file/data

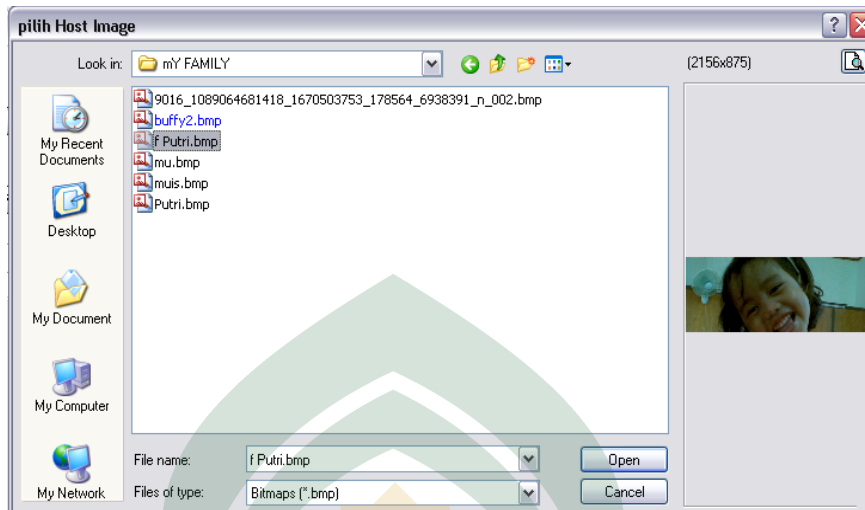
c. Menu Dekripsi (Proses Pengungkapan)

Menu proses Pengungkapan Pesan akan ditampilkan pada saat pengguna memilih Menu “Dekripsi”, pada menu beranda Dekripsi ada tombol Proses Pengungkapan, dapat dilihat Gambar V.6. berikut :



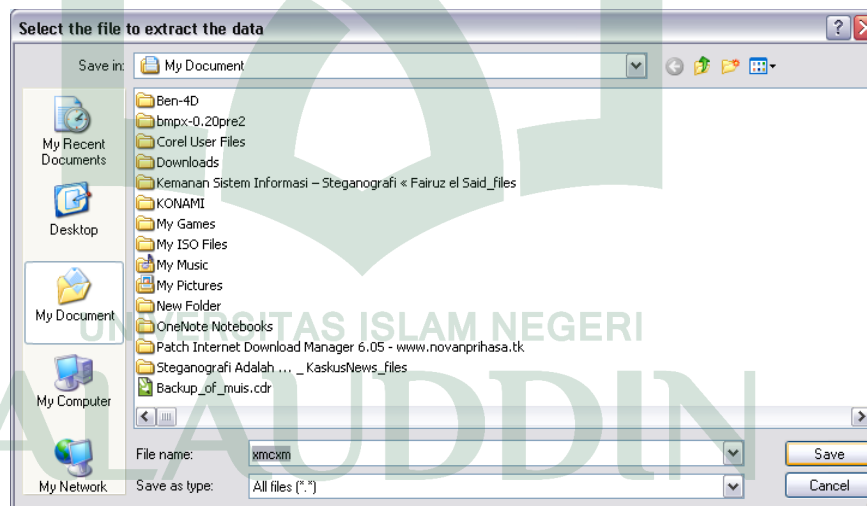
Gambar V.6 Menu proses dekripsi pesan

jika tombol dipilih pada aplikasi StegoBitmap. akan menampilkan *Dialog* untuk mengambil kembali *Image* yang telah disisipi pesan dan *Dialog* untuk menyimpan hasil ekstrak. implementasi untuk memilih *Image* yang telah disisipi pesan dapat dilihat pada Gambar V.7. berikut :



Gambar V.7. Dialog memilih hasil stego

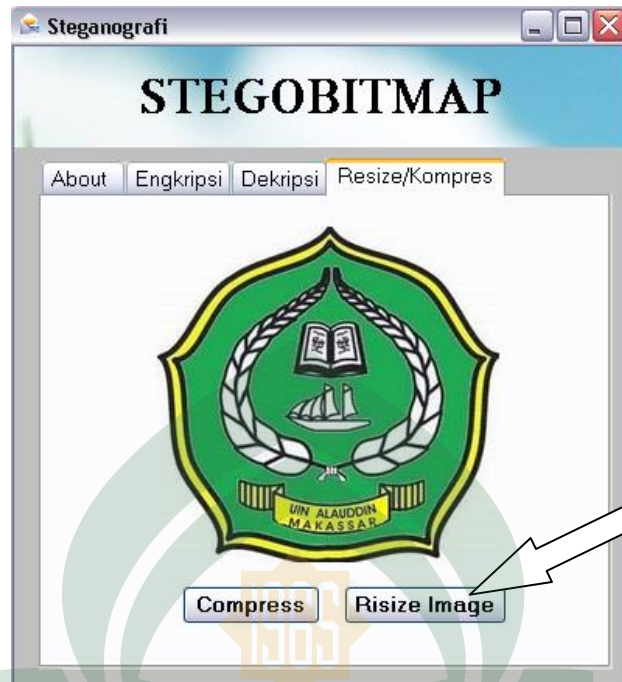
Kemudian akan muncul *Dialog* untuk menyimpan hasil pengungkapan, implementasiya dapat dilihat pada Gambar V.8. berikut :



Gambar V.8 Dialog memilih lokasi dekripsi

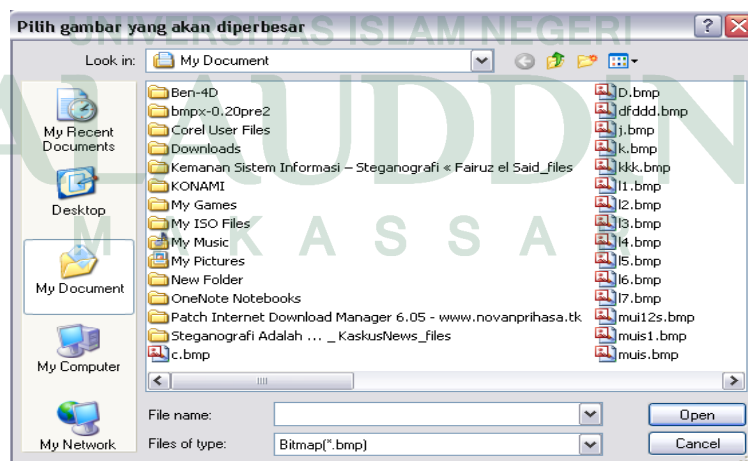
d. Menu Proses Resize

Proses resize akan ditampilkan pada saat pengguna memilih tombol tab resize/kompres. Lihat Gambar V.9. berikut :



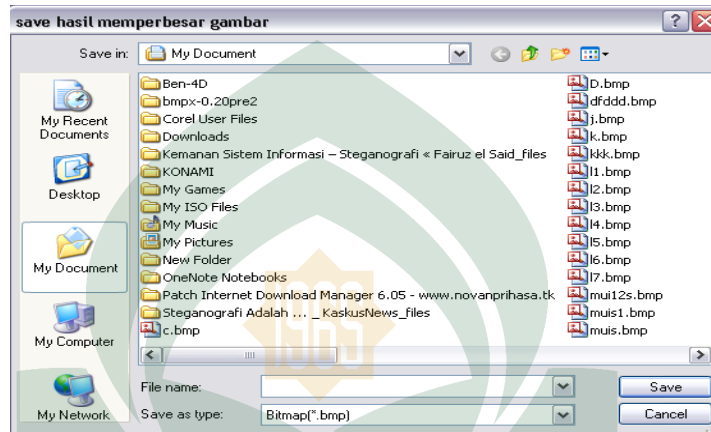
Gambar V.9 Menu resize

“Resize”. Pada proses ini akan menampilkan *Dialog* untuk memilih *Image* yang akan di perbesar pixelnya dan menyimpan hasilnya. Pada *Dialog* untuk memilih *Image* akan diimplementasikan pada Gambar V.10. berikut :



Gambar V.10 Dialog memilih resize image (yang diperbesar)

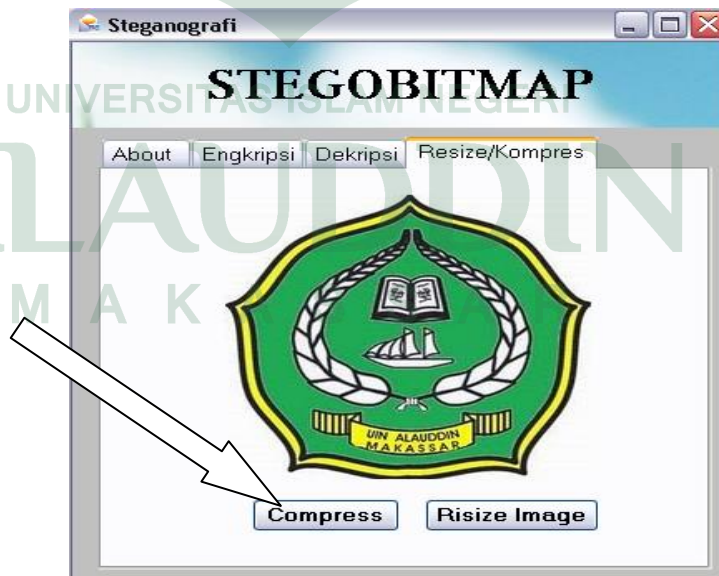
Selanjutnya akan tampil *Dialog* untuk penyimpanan hasil pembesar *Image*. Implementasinya dapat dilihat pada Gambar V.11. sebagai berikut:



Gambar V.11 *Dialog menyimpan hasil resize*

e. Menu Proses Kompres *File*

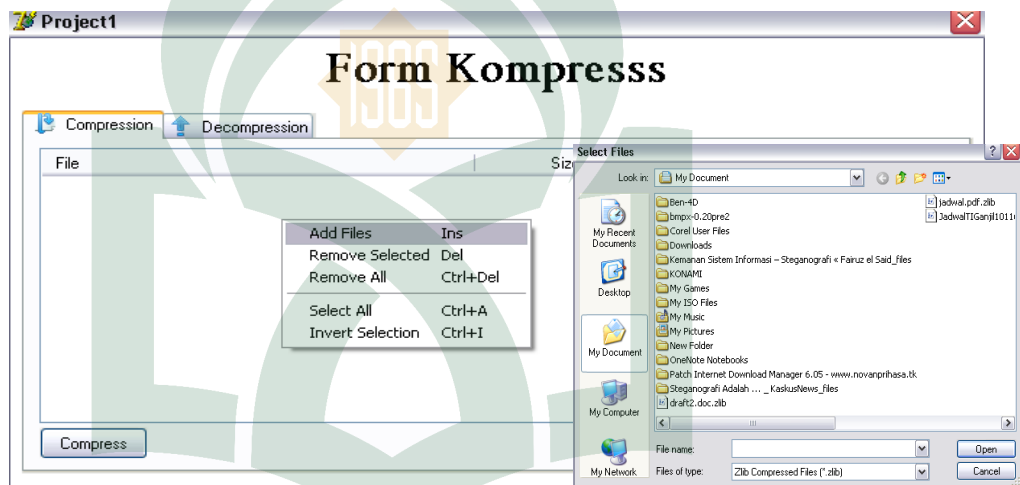
Menu proses Kompres *file* akan ditampilkan pada saat pengguna memilih tam Resize/ kompres. Dapat dilihat Gambar V.12. berikut :



Gambar V.12 *Menu kompres*

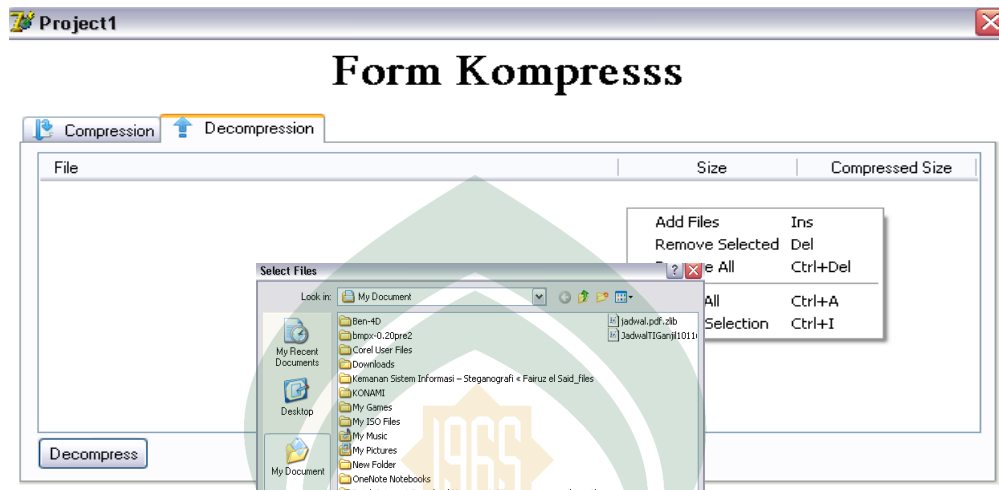
tombol “Kompres”. Pada menu ini akan ada dua proses yaitu proses kompres (mengurangi kapasitas *file*) dan Decompres (mengembalikan kapasitas *file*).

Pada pilih proses Compres dapat dilakukan dengan memilih *file* dengan cara klik kanan *Add file*, maka akan muncul *Dialog* untuk memilih *file*. implementasinya dapat dilihat pada Gambar V.13. sebagai berikut :



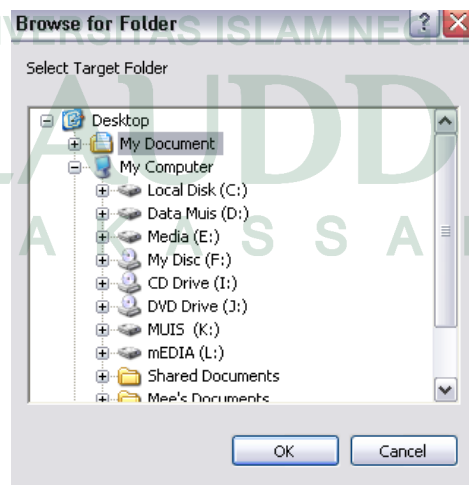
Gambar V.13 Proses Kompres file

Pada pilih proses Decompres dapat dilakukan dengan memilih *file* dengan cara klik kanan *Add file* maka akan muncul *Dialog* untuk memilih *file* sudah di Compres, implementasinya dapat dilihat pada Gambar V.14. sebagai berikut :



Gambar V.14 Proses dekompres file

Setelah *file* sudah di pilih maka tekan tombol Compres dan Decompres, maka akan muncul *Dialog* untuk memilih lokasi hasil kompres dan decompres, implementasinya dapat dilihat pada Gambar.V.15. berikut :



Gambar V.15. Dialog menyimpan hasil kompres dan dekompres

B. Pengujian Sistem

Pada tugas akhir ini, pengujian perangkat lunak dilakukan dengan pengujian *White Box*. Pengujian *White Box* merupakan metode desain uji kasus yang menggunakan struktur *control* dan desain *prosedural* untuk menghasilkan kasus-kasus uji. Pengujian perangkat lunak steganografi ini mencakup, proses penyisipan dan ekstraksi pesan, serta kinerja perangkat lunak lainnya, yaitu dengan membandingkan kualitas gambar sebelum dan setelah penyisipan.

1. Lingkungan Pengujian

Pengujian dilakukan pada lingkungan yang sama dengan lingkungan implementasi. Dikarenakan aplikasi yang dibangun merupakan aplikasi *Desktop based*. Proses pengujian perbandingan kualitas juga dilakukan menggunakan *notebook* yang sama dengan saat implementasi.

2. Tujuan Pengujian

Tujuan dari pengujian program yang dilakukan :

- a. Menguji proses Kompres dan *Resize Image*
- b. Menguji proses penyisipan dan ekstraksi pesan
- c. Menguji dampak perubahan gambar setelah proses penyisipan
- d. Menguji seberapa besar *file* yang bisa ditampung oleh *Image* yang mempunyai pixel 1024 x 768.

3. Data Uji

File gambar yang akan digunakan pada pengujian : “Hutan.bmp”, dengan resolusi 1024 x 768.

4. Kasus Uji

Berdasarkan tujuan pengujian yang telah didefinisikan sebelumnya, maka terdapat tiga buah kasus untuk kebenaran perangkat lunak, yaitu kasus uji satu hingga tiga. Sedangkan pengujian kinerja perangkat lunak akan diuji pada kasus uji 4.

a. Kasus uji 1

Pengujian ini dilakukan untuk menguji kebenaran validasi format gambar sebagai masukan pada proses enkripsi dan proses Dekripsi. Cara yang dilakukan adalah memasukkan format gambar bmp, dimana pengujian dinyatakan berhasil jika proses penyisipan dapat dilakukan pada gambar dengan format BMP.

b. Kasus Uji 2

Pengujian ini dilakukan dengan cara menyisipkan pesan ke dalam gambar, kemudian mengekstraksinya kembali. Gambar yang menjadi media pada pengujian ini adalah gambar yang dinyatakan valid dari hasil kasus uji.

c. Kasus Uji 3

Pengujian ini dilakukan untuk menguji kemampuan kompres dan resize image dari perangkat lunak, yang akan dikompres adalah dari *file*/data yang berbeda.

d. Kasus Uji 4

Pengujian ini dilakukan untuk menguji data yang sebelumnya lebih besar dari pada wadah, sehingga akan melalui melalui teknik kompres data sehingga data tersebut bisa ditampung

e. Kasus Uji 5

Pengujian ini dilakukan untuk menguji kualitas dari gambar yang dihasilkan setelah melalui proses penyisipan, yaitu dengan membandingkannya dengan gambar yang asli. Dimana perbandingan dari dua gambar hanya dilakukan secara subjektif (gambar dianggap mirip)

f. Kasus Uji 6

Pengujian ini dilakukan dengan menguji Ratio antara ukuran *file* dan *Image* yang bisa ditampung.

g. Kasus Uji 7

Pengujian ini dilakukan dengan mengirim *Image* yang sudah disisipkan *file* melalui *Email* dan *Facebook*. Dan melakukan pengungkapan (ekstrak) kembali terhadap *Image* yang dikirim.

C. Analisis dan Hasil Perangkat Lunak Steganografi

Berikut ini adalah hasil pengujian dari kasus uji yang dilakukan, beserta evaluasi dari masing-masing kasus uji.

1. Kasus uji 1

Pengujian ini dilakukan untuk menguji kebenaran validasi format gambar sebagai masukan pada proses penyisipan dan ekstraksi. Cara yang dilakukan adalah memasukkan format gambar bmp, dimana pengujian dinyatakan berhasil jika proses penyisipan dapat dilakukan pada gambar dengan format BMP.

Tabel V.2 Hasil Pengujian Kasus Uji 1

File Gambar	Format	Hasil Validasi	Kesimpulan
“pemandangan”	BMP	Valid	sukses
“Hutan 06”	BMP	Valid	Sukses
“hutan”	BMP	Valid	Sukses

Dari hasil tersebut, terbukti bahwa StegoBitmap telah berhasil menjalankan fungsi *parsing* gambar BMP, sehingga validasi format gambar atau pengambilan nilai dalam gambar dapat dilakukan dengan baik.

2. Kasus uji 2

Pengujian ini dilakukan dengan cara menyisipkan *file* ke dalam gambar, kemudian mengekstraksinya kembali. *File* yang digunakan sebagai bahan percobaan adalah *file yang berbeda*. Gambar yang menjadi media pada pengujian ini adalah gambar yang dinyatakan valid dari hasil kasus uji 1.

Hasil penyisipan pesan ini ditunjukkan pada tabel V.3. untuk hasil pengujian dari gambar asli dengan gambar yang telah disisipkan pesan dapat dilihat pada hasil pengujian uji kasus 4.

Tabel V.3 Hasil Pengujian Kasus Uji 2 (Penyisipan)

Masukan Gambar	File	Keluaran Gambar	keterangan
“Hutan.bmp” (1024X768)	“coba.doc”(136kb)	“coba.bmp”	Diterima
	“coba.exe”(82.0 kb)	“coba2.bmp”	Diterima
	“cobaT.mp3”(952 kb)	X	<i>File</i> tidak muat
	“Gunung.jpg” (107 kb)	“Sampah”	Diterima

Setelah proses penyisipan selesai, dilakukan proses ekstraksi dari masing-masing gambar. Hasil dari proses ekstraksi dapat dilihat pada tabel V.4 dimana *file* coba.mp3.zlib tidak mampu melakukan penyisipan pada *Image* hutan.bmp, karna kemampuan gambar lebih kecil dari *file* yang akan dilakukan proses penyisipan sehingga menyebabkan adanya pesan error. sedangkan yang lainnya mampu menampung *file* tersebut. yang digunakan tetap sama untuk menghasilkan proses ekstrak yang sempurna atau valid.

Tabel V.4 Hasil Pengujian Kasus Uji 2 (Pengungkapan)

Keluaran gambar	Decompress	Keluaran	kesimpulan
“coba.doc.bmp”	Coba, (136 kb)	“coba” isi sama	Sukses
“coba.exe.bmp”,	Coba2, 122 kb	“coba2” isi sama	Sukses
“sampah.bmp”	Tdk terkompres	“Gunung.jpg”	Kualitas sama

Dari pengujian ini, terbukti bahwa StegoBitmap sudah berhasil menjalankan proses penyisipan dan ekstraksi gambar dengan benar. Semua pesan yang menjadi masukan telah berhasil disisipkan, dan kemudian dapat diekstraksi kembali dengan baik. Pesan yang diekstraksi sama dengan pesan yang asli.

3. Kasus uji 3

Pengujian ini dilakukan untuk menguji kemampuan kompres dan resize image dari perangkat lunak, yang akan dikompres adalah dari *file/data* yang berbeda. Hasil dari kompres dan resize bisa dilihat pada Gambar.V.5 dan V.6. sebagai berikut :

Tabel V.5 Hasil Pengujian Kasus Uji 3 (Kompres)

File/data	Hasil compress	Ratio	kesimpulan
“coba.doc”(523 kb)	Coba, (34kb)	10%	sukses
“coba.exe”(82 kb)	Coba2, (60 kb)	72%	sukses
“cobaT.mp3”(953kb)	cobaT.mp3.zlib(890kb)	93%	sukses

Pada kasus percobaan diatas,disimpulkan bahwa ketiga *file* yang berbeda ini memiliki ratio compres yang beda.

Tabel V.6 Hasil Pengujian Kasus Uji 3 (resize)

Gambar	Hasil Resize	kesimpulan
“hutan”(2.25 Mb) (1024X768)	(2048X1536), 9.00MB	Sukses
“hutan 06”(1.83 Mb) (800X600)	(1600X1200), 5.49MB	Sukses
“pemandangan”(1.17 Mb) (640X480)	(1280X960), 3.51MB	Sukses

4. Kasus uji 4

Pengujian ini dilakukan untuk menguji data yang sebelumnya lebih besar dari pada wadah, sehingga akan melalui melalui teknik kompres data sehingga data tersebut bisa ditampung. Hasil pengujian dapat dilihat pada table V.7.dan V.8. Sebagai berikut :

Table V.7 Hasil Pengujian Kasus Uji 4 Sebelum di kompres

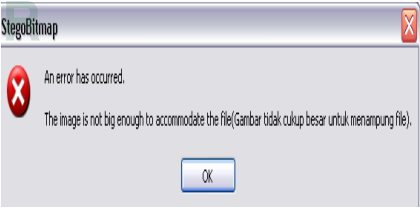
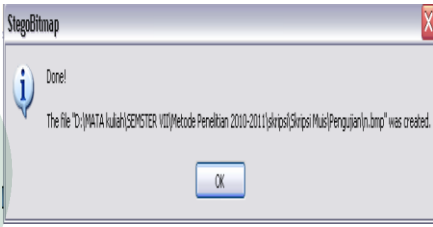
Ukuran data	Gambar	Hasil
Coba11.doc (1.12 Mb)	Hutan”(2.25 Mb) (1024X768)	Akan terjadi error bahwa <i>file</i> yang ditampung tidak muat 

Table V.8 Hasil Pengujian Kasus Uji 4 Setelah di Kompres

Ukuran data	Gambar	Hasil
coba3.doc (801 Kb) setelah dikompres ukurannya berubah menjadi 49 kb.	Hutan”(2.25 Mb) (1024X768)	Maka data tadi yang akan ditampung bisa di proses 







Pada kasus uji 5 ini disimpulkan bahwa, data lebih besar dari wadah. Maka akan terjadi pesan error “bahwa gambar muat untuk menampung data”. Sehingga data tersebut dikompres. Sehingga kapasitas data tersebut bisa diperkecil, dan pada saat dilakukan proses penyisipan, maka data tersebut bisa ditampung.

5. Kasus uji 5

Pengujian ini dilakukan dengan cara membandingkan gambar hasil pengujian kasus uji 2 untuk penyisipan dengan *file* gambar yang asli secara subjektif.

Hasil dari pengujian kasus uji 4 dapat dilihat pada tabel V.9. sebagai berikut:

Tabel V.9. Hasil Pengujian Kasus Uji 5 Untuk kualitas Gambar setelah dan sebelum disisipkan Pesan



Ukuran Pesan yang disisipkan	File Gambar Asli	File Gambar Setelah Disisipkan
801 kb		
850 kb		
200 kb		

Pada percobaan diatas maka semua citra hasil penyisipan diaggap mirip dengan citra aslinya. Sehingga StegoBitmap dapat menyisipkan pesan tanpa menimbulkan kecurigaan atau secara kasat mata.

6. Kasus uji 6

Pengujian ini dilakukan dengan mengukur batas *file* yang bisa tampung oleh *image*. dari pengujian kasus uji 6 dapat dilihat pada tabel V.10. Sebagai berikut :

Tabel V.10. Hasil Pengujian Kasus Uji 6 Untuk Ratio ukuran *file* dan gambar



Ukuran Pesan yang disisipkan	File Gambar Asli	File Gambar Setelah Disisipkan
896 kb	1024x768 	Error
850 kb	1024x768 	Berhasil

Kesimpulan dari kasus uji 6 ini adalah pada pixel 1024x768 ini hanya bisa menampung *file* dengan ukuran maksimum 850 kb. Jika lebih dari ukuran *file* tersebut bisa di pakai alternative yaitu Resize *Image* yaitu memperbesar *Pixel* nya dan Kompres *file*.

7. Kasus Uji 7

Pengujian ini dilakukan dengan megirim *Image* yang sudah disisipkan *file* melalui *Email* dan *Facebook*. dari pengujian kasus uji 6 dapat dilihat pada tabel V.11. sebagai berikut :

Tabel V.11. Hasil Pengujian Kasus Uji 7 Untuk pengiriman gambar yang sudah disisipi Pesan Lewat via E-mail.

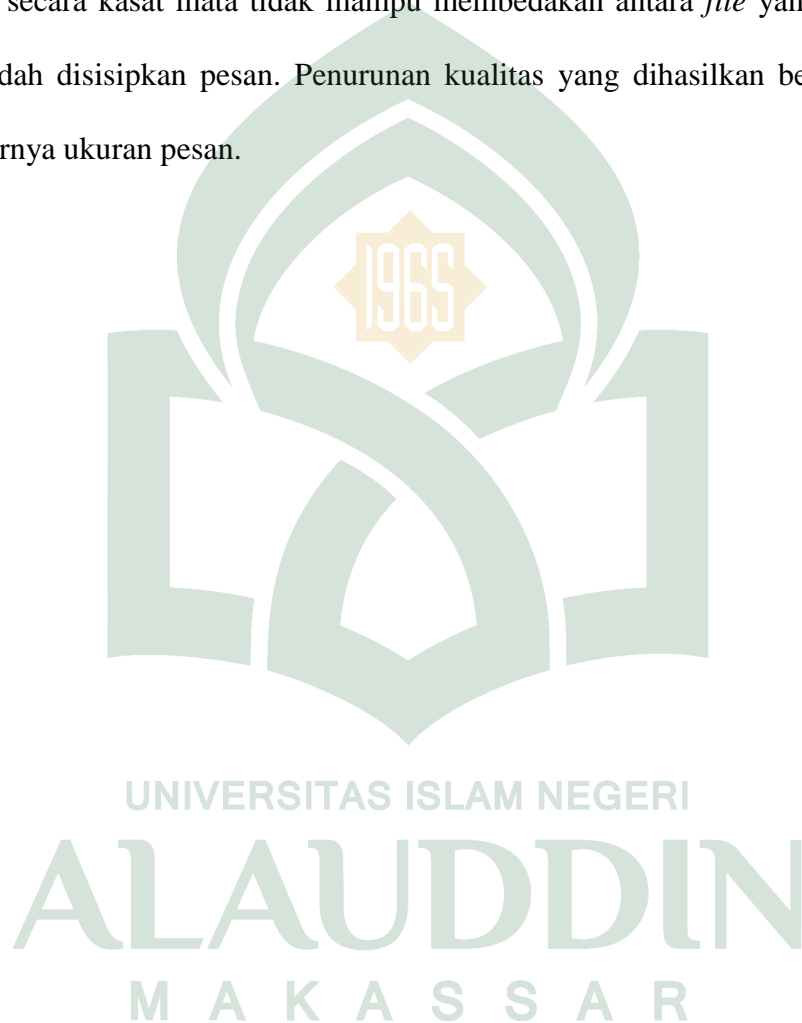
File Gambar Asli yang dikirim	Pengiriman lewat	Pengungkapan
1024x768 	E-mail (sukses)	Sukses
1024x768 	Facebook (sukses)	Berhasil

Pada pengujian ini dapat disimpulkan bahwa pada saat di lakukan pengungkapan dari hasil pengiriman ini bisa dikatakan berhasil dengan baik. Karna bisa dilakukan pengungkapan kembali *file* yang dikirim.

Dari semua hasil pengujian yang telah dilakukan, perangkat lunak StegoBitmap dapat menjalankan semua fungsi dan spesifikasi dengan benar, Ukuran *file* yang disisipkan terbatas terhadap image yang akan disisipi, sehingga ukuran maksimalnya adalah 850 kb dengan ukuran pixel 1024 x 768, jika melampaui batas yang maksimal ukuran *file* tersebut maka akan terjadi error sehingga alternatifnya yang bisa dilakukan adalah melakukan Kompres *file* (memperkecil ukuran *file*) dan Resize Image (mempebesar ukuran pixel dua kali dari ukuran aslinya) sehingga dapat menutupi kelemahan dari aplikasi dari

metode LSB ini. Penggunaannya dalam hal media pengiriman bisa dilakukan di media internet sesuai pengujian yang telah dilakukan.

penurunan kualitas yang dihasilkan pada gambar tidak terlalu signifikan. Dan secara kasat mata tidak mampu membedakan antara *file* yang sebelum dan sesudah disisipkan pesan. Penurunan kualitas yang dihasilkan bergantung pada besarnya ukuran pesan.



BAB VI

PENUTUP

A. Kesimpulan

Dari kegiatan-kegiatan yang telah dilakukan terkait dengan pelaksanaan tugas akhir, dapat disimpulkan bahwa:

1. Telah berhasil dikembangkan perangkat lunak yang dapat melakukan steganografi pada gambar dengan format *Bitmap*. Kebutuhan fungsional dari perangkat lunak, seperti proses penyisipan dan ekstraksi pesan sudah dapat dilakukan dengan benar.
2. Maksimal ukuran *file* yang dapat disembunyikan adalah 850 kb pada Image yang mempunyai pixel 1024 x 768.
3. Teknik Steganografi ini mampu menyimpan data yang lebih besar dari pada kapasitas maksimum wadah sebenarnya. Dan dari hasil pengujian ini berhasil menyimpan data berukuran apapun, hal ini mengingat kemungkinan untuk memperbesar wadah secara dinamis sesuai ukuran data dan memperkecil kapasitas data dengan jalan kompres.

4. Metode *Least Significant Bit* sebagai metode penyisipan pesan sudah dapat dilakukan dengan benar yaitu menyisipkan pesan rahasia dalam *pixel* yang tak signifikan, dari *Image*.
5. Kualitas *Image* dan besar *size* yang dimiliki *Image* setelah proses penyisipan sama dengan kualitas *Image* dan *size*nya sebelum dilakukan penyisipan sehingga, penyisipan ini berhasil dilakukan tergantung dari besar kecilnya ukuran pesan yang disisipkan.

B. Saran

Adapun saran terkait dengan pelaksanaan tugas akhir ini, terutama peneliti yang tertarik dengan steganografi untuk menambahkan dukungan fungsi-fungsi untuk pengembangan aplikasi steganografi.

DAFTAR PUSTAKA

- Departemen Agama RI, Alquran dan terjemahannya, Semarang: Toha Putra, 1989.
- Davis, William S, Sistem Analisis and Design : A Structured Approach, United State of America: Addison-Westley Publishing Company.
- Ferry Pangaribuan, Aplikasi Penyembunyian Pesan Metode MARS Metode dan Zhang LSB Image. Bandung : Institut Teknologi Bandung, 2008.
- Gonzales, R.C., Woods, R.E. Digital Image Processing, Addison-Wesley Publishing Company, 1992.
- Hernawan Sulistyanto., Kompresi Data Lossless dengan Metode Lempel-Zip, Teknik Elektro Universitas Muhammadiyah Surakarta, 2003.
- Muhammad Hakim, Implementasi Penyembuyian pesan dengan metode LSB, Bandung : Institut Teknologi Bandung, 2009.
- Muhamad Firdaus, penentuan kombinasi teknik kompresi untuk mendukung penyimpanan data akademik pada smartcard, Institut Teknologi Sepuluh Nopember.2009.
- Morkel, T., Eloff, J.H.P., Olivier, M.S. An Overview of Citra Steganography, 2005.
- Prasetyo Andy Wicaksono, Penyembunyian Pesan pada Citra GIF Menggunakan Metode Adaptif, Bandung : Institut Teknologi Bandung, 2009.

Rinaldi Munir, Pengolahan Citra Digital dengan Pendekatan Algoritmik, Bandung : Informatika, 2004.

Ronald Augustinus Penalosa, *Steganografi Pada Citra dengan Format GIF Menggunakan Algoritma GifShuffle*, Bandung: ITB, 2008.

Suarga, M. Sc., M. Math., Ph. D., Algoritma Pemograman, Makassar : 2004.

TIK, *Mengenal program Grafis*, Yokyakarta :SMA Negeri 1 Yokyakarta. 2008.

Winda Winanti, Penyembuyian pesan pada citra terkompresi JPEG menggunakan metode Spread Spectrum, Bandung : Institut Teknologi Bandung, 2009.

Willy Sudiarto Raharjo, Aditya Wikan Mahastama *Permodelan System Perangkat Lunak, Uses Case UML*, Univ Kristen Duta Wacana, PSPL.

Westfeld, A., Pfitzmann, : *Attacks On Steganographic System*, 1999.

Yulie Anniera Sinaga. Program Steganalisis Metode LSB pada Citra dengan Enhanced LSB, Uji Chi-Square, dan RS-Analysis, Bandung : Institut Teknologi Bandung, 2009.





UNIVERSITAS ISLAM NEGERI
ALAUDDIN
M A K A S S A R